



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1978-09

High speed data acquisition system.

Elliott, Mack Taylor

Monterey, California: U.S. Naval Postgraduate School

<http://hdl.handle.net/10945/28263>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



<http://www.nps.edu/library>

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

HIGH SPEED DATA ACQUISITION SYSTEM

Mack Taylor Elliott

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

HIGH SPEED DATA ACQUISITION SYSTEM

by

Mack Taylor Elliott

September 1978

Thesis Advisor:

Louis V. Schmidt

Approved for public release; distribution unlimited

T185389

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) HIGH SPEED DATA ACQUISITION SYSTEM		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; Sept 1978
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Mack Taylor Elliott		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1978
		13. NUMBER OF PAGES 191
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) analog to digital converter data acquisition microcomputer microprocessor unsteady aerodynamics		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This paper describes the expansion and modification of an existing data acquisition system to effect extensive improvements in speed and flexibility. A microprocessor, flexible disk drive, analog to digital converter, direct memory access module, and high-speed line printer were integrated and interfaced to an IBM 360 digital computer with a high-speed data transmission line. The resultant system provided the capability of digitizing up to		

sixteen analog inputs simultaneously at rates in excess of 45,000 samples per second. The experimental data could be transmitted expeditiously to the IBM 360 computer for efficient manipulation. Additional benefits gained from the system were its capabilities as a remote terminal for the IBM 360 and a typewriter-quality word processor. The data acquisition and reduction system was qualified for functional performance and speed through a series of test exercises. The word processor was demonstrated in the production of this document.

HIGH SPEED DATA ACQUISITION SYSTEM

by

Mack Taylor Elliott
Lieutenant, United States Navy
B. S., University of South Carolina 1971

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

September 1978

ABSTRACT

This paper describes the expansion and modification of an existing data acquisition system to effect extensive improvements in speed and flexibility. A microprocessor, flexible disk drive, analog to digital converter, direct memory access module, and high-speed line printer were integrated and interfaced to an IBM 360 digital computer with a high-speed data transmission line.

The resultant system provided the capability of digitizing up to sixteen analog inputs simultaneously at rates in excess of 45,000 samples per second. The experimental data could be transmitted expeditiously to the IBM 360 computer for efficient manipulation. Additional benefits gained from the system were its capabilities as a remote terminal for the IBM 360 and a typewriter-quality word processor. The data acquisition and reduction system was qualified for functional performance and speed through a series of test exercises. The word processor was demonstrated in the production of this document.

TABLE OF CONTENTS

I.	Introduction.....	10
II.	Hardware.....	12
A.	Components.....	13
1.	Microprocessor.....	14
2.	Analog to Digital Converter.....	14
3.	Direct Memory Access.....	16
4.	High-speed Printer.....	17
5.	Full-sized Digital Computer.....	17
B.	Interfaces.....	18
1.	Printer Interface.....	19
2.	High-speed Line Interface.....	20
3.	Analog to Digital Converter Interface.....	21
4.	Direct Memory Access Interface.....	22
III.	Software.....	23
A.	PRINT Program.....	25
1.	Printer Control.....	26
2.	File Reading.....	27
3.	Formatting.....	27
4.	Prompts.....	27
5.	PRINT User's Guide.....	28
B.	LINK Program.....	28
1.	USART Setup.....	30
2.	Monitor Function.....	30
3.	Data Buffers.....	31

4. LINK User's Guide.....	33
C. GO Program.....	33
1. Data File Parameters.....	35
2. ST-800 and DMA Setup.....	36
3. DMA Reset.....	36
4. GO User's Guide.....	37
D. DATLINK Program.....	37
IV. System Qualification.....	39
A. Shannon's Sampling Theorem.....	39
B. Qualification Test.....	39
C. Data Sampling Theory.....	40
D. Fourier Analysis.....	41
E. Interchannel Sampling Delay.....	42
F. REDUCE Fourier Coefficient Program.....	43
G. System Qualification Results.....	43
V. Alternative Solutions.....	45
A. Dual-Interrupt Data Acquisition.....	45
B. Model 40 Printer as List Device.....	48
VI. Conclusions.....	50
A. Future System Improvements.....	50
APPENDIX A: Glossary.....	52
APPENDIX B: GO User's Guide.....	55
APPENDIX C: LINK User's Guide.....	59
APPENDIX D: DATLINK User's Guide.....	65
APPENDIX E: PRINT User's Guide.....	67
APPENDIX F: GO Program Listing.....	70
APPENDIX G: LINK Program Listing.....	88
APPENDIX H: DATLINK Program Listing.....	116

APPENDIX I: PRINT Program Listing..... 138

APPENDIX J: GO2 Program Listing..... 157

APPENDIX K: Patch to CP/M BIOS..... 176

APPENDIX L: ON Program Listing..... 177

APPENDIX M: REDUCE Fortran Listing..... 179

APPENDIX N: Reduced Data File Listings..... 184

LIST OF REFERENCES..... 190

INITIAL DISTRIBUTION LIST..... 191

LIST OF FIGURES..... 8

LIST OF FIGURES

1. Handshaking on the Model 40 Printer.....	20
2. Handshaking on the high-speed line.....	21
3. Dual-interrupt timing.....	47

ACKNOWLEDGEMENTS

The author wishes to express his gratitude to the many individuals who contributed their time and efforts to this project. LCDR Duane Englehardt and Ted Dunton, both of the Department of Aeronautics, and Kathryn Strutynski of the W. R. Church Computer Center furnished untiring assistance with technical problems and programming. Professor Louis V. Schmidt of the Department of Aeronautics provided guidance and expertise during all phases of development. Also, a very special thanks is due to Betty Ann, Jeremy, and Joshua for their support and infinite patience.

I. INTRODUCTION

The advent of the low-cost microprocessor system has made possible the conduct of numerically controlled laboratory experiments such as described by Casko, Ref. 11. An equally important application is in data acquisition and analysis. The ability of the microprocessor to accommodate many different tasks by software (program) changes has resulted in a very flexible system for an academic laboratory environment. Because of the relatively low cost of a complete microprocessor system, which should more aptly be termed a microcomputer, it is now practical to do experiments in aeronautics with an orientation toward investigating unsteady or time varying physical behavior. Recent experiments on the Circulation Controlled Airfoil, as reported by Englehardt in Ref. 1, are an example of the type of work which can be done economically in establishing the frequency response behavior of aerodynamic configurations.

In improving the experimental capabilities of the microcomputer system in the Department of Aeronautics, several features became evident as desired goals. Included in these goals were:

- A. To extend the useful frequency range for data acquisition by verifying the Analog to Digital (A/D) sampling rate potential of an existing system data card as being on the order of 40,000 samples per second for situations of routine usage.
- B. To upgrade the use of output printing devices to a typewriter-quality line printer with a maximum output baud rate of 9600 in comparison to the more common Teletype Model ASR-33 or ASR-35 baud rate of 110.
- C. Although the microcomputer system had an internal computational package allowing the option of software programming for data reduction in BASIC language, it was desirable to link the microcomputer system to the IBM 360/67 digital computer at the W. R. Church Computer Center for increasing the scope (both complexity and speed) of data reduction for digitized data sets.

This thesis describes the approaches taken to achieve the above stated goals in order to improve both system flexibility and computational speed while retaining the advantages of local autonomy and cost effectiveness provided by the use of a microcomputer system.

II. HARDWARE

The original concept of the microcomputer or micro-processor involved the design of a low-cost compact version of the large digital computers. According to Osborne, Ref. 12, the resultant design differed from the goal primarily due to the distribution of logic on integrated circuit chips. Some differences in addressing modes and execution times were evident in the microcomputers.

The system used in this project had a sixteen line address bus capable of addressing 65,536 locations (2 to the 16th power). Data processed by the microcomputer travelled over an eight line data bus. The data bus is capable of handling eight binary digits (bits), or one byte, at a time. Similarly the central processor unit (CPU) within the microcomputer can work with only one byte at a time. Although sixteen bit CPU's and data busses have recently been developed, the large number of existing eight bit CPU chips assures us that the eight bit bus will be in usage for quite some time.

Subsequently data processing or numerical manipulation in the eight bit system is a relatively slow and pedestrian process. Numerical accuracy requires representing a number

by several bytes, and in much of our software the floating point binary number is represented by four bytes consisting of exponent, sign, and magnitude. Long cumbersome algorithms manipulate one byte at a time and then collocate the individual results into one total number. The addition of a peripheral device specializing in numerical manipulation, called a "math pack", can expedite the process considerably. However, all input/output operations would still be limited by the eight binary parallel digit capacity of the CPU and data bus.

The approach taken in this thesis was to avoid, to the greatest extent possible, any data manipulation by the microprocessor and instead to use it only as a control for faster peripheral devices. The data manipulation was then accomplished with the IBM 360 digital computer.

A. Components

The major components utilized in the project are discussed briefly in this thesis, and detailed descriptions are given in the referenced material. Because of the inherent complexity of integrated circuitry and digital logic considerations, even the reference manuals are often incomplete. Ignorance of a subtle but important detail about a particular component can cause the neophyte student of microprocessor technology to make errors which are

difficult to identify and cause unpredictable results.

Emphasis has been put, therefore, on identifying particular idiosyncracies which have been exposed during this project and hopefully the errors need not be repeated.

1. Microprocessor

The Intel MDS-800 Microcomputer Development System with central processor unit, 64K of random access memory, front panel controller, and mainframe enclosure has been documented extensively in Ref. 1. The MDS-800 and connected flexible disk drives, CRT terminal, and paper tape reader were the benchmark devices for the project. The system, although not quite state-of-the-art in terms of micro-processors, was nevertheless a well-developed and popular system for which substantial software had been developed.

2. Analog to Digital Converter

The Datal Sinetrac-800 Analog to Digital Converter, also described in Ref. 1, was reconfigured according to the specifications in Ref. 2 for use in the Direct Memory Access (DMA) mode. Basically the only changes necessary were disabling the address structure to prevent the CPU from writing to the converter directly, and enabling the circuit board for DMA operation. Parameters left unchanged included the input voltage range of +/- five volts, twelve bit reso-

lution, twos complement output coding with sign extension, and the scan-clock option enabled. The converter digitized each analog signal into two bytes which required two memory locations. The least significant twelve bits provided a resolution of two to the 12th power (4096). When applied to the input voltage range, this resolution meant an accuracy of ± 0.002 volts. The remaining four bits of the digitized input formed a hex digit, either 0 or F, which represented a positive or negative sign. Connection of the external analog inputs to the converter was made via a locally prepared terminal box.

Several options were available for determining the scan repetition rate. The scan-clock option allowed for a hardware variable scan rate but did not provide enough flexibility. Another possibility was to use software control through the CPU but this option was too slow. An approach which provided a greater degree of flexibility utilized the SBC Intel 534 Input/Output board to time the scan intervals, and involved operating the ST-800 on an interrupt basis so the interrupt structure was enabled. The final configuration, however, excluded interrupts by the device, hence the interrupt logic wiring was again disabled.

3. Direct Memory Access

The Intel SBC-501 Direct Memory Access (DMA) Channel

Controller board was utilized to greatly decrease the throughput time of analog signal to memory storage. As reported in Ref. 1, the analog to digital converter, when operated under direct program control, had a throughput time of 76.5 microseconds per channel. This relatively slow rate was caused by the necessity of multiple transfers of each word of converted data from converter to CPU to memory with each transfer requiring several time-consuming commands to be issued by the CPU.

According to the specifications in Ref. 3, the DMA controller board was configured for base address and interrupt level and installed in the MDS-800 mainframe. A wiring harness obtained from the Datel Corporation connected the DMA board to the ST-800 converter. The DMA was programmed by the CPU to transfer a specific number of data words from the converter directly to random access memory. Then control of the data bus was relinquished by the CPU and the DMA and ST-800 were allowed to work together at maximum speed. Using full handshaking to avoid data overruns, the ST-800 sampled and converted analog signals which were routed through the DMA directly into memory. The CPU was bypassed and consequently the throughput time was reduced to 21.7 microseconds. Utilization of a pulse generator to initiate each scan gave total flexibility to the data sampling rate within the outside limit of 45,000 Hertz.

4. High-speed Printer

The Teletype Model 40 Printer was chosen to supplement the teletype terminal used in earlier projects. The Model 40 is a chain-type printer capable of 9600 baud (or 960 characters per second). Upper and lower case letters are available as is the option to use a variety of paper sizes. The printer was interfaced through a serial transmission Universal Synchronous Asynchronous Receiver Transmitter (USART) on the Intel SBC 534 board and programmed to use the standard 11 X 14 inch paper stock. Switch selectable options on the printer were set as desired in accordance with Ref. 4. The major problem that occurred when interfacing the printer was an incorrectly wired interconnector in the printer enclosure.

5. Full-sized Digital Computer

The International Business Machines Model 360/67, located in the W. R. Church Computer Center, was interfaced to the microprocessor via an RS-232C driver and telephone line. The interface, called a "high-speed line" because of its improved speed of transmission over earlier connections, was also serially driven by a USART on the SBC 534 board. Operating at baud rate of 1200 baud, the interface provided the capability of transmitting data to the larger computer which was designed for more efficient data manipulation.

The line from the microprocessor fed into the IBM 360 through an IBM 2701 Data Adapter unit controlled by the Control Program-67/Cambridge Monitoring System. Interface requirements that were imposed by the IBM 2701 were obtained from Ref. 5.

B. Interfaces

The Intel SBC 534 Four Channel Communications Expansion Board, described in Ref. 6, was used to interface the microprocessor with both the printer and the high-speed line. The SBC 534 board was selected because of the flexibility it afforded with regard to future improvements to the system. The board was jumper configured for base address, installed in the MDS-800 mainframe, and connected to the high-speed line and printer by locally prepared wiring harnesses. Two of four serial 8251 USART's and two of six programmable timer circuits on the board were utilized for the interfaces. One Programmable Interrupt Controller (PIC) of two on the board was used in an alternate approach mentioned later, but the final configuration left the PIC disabled. Another circuit available on the board for future use is an 8255 Programmable Peripheral Interface. Exact specifications and operational descriptions of the individual circuits on the SBC 534 board were found in Refs. 7 and 8.

The rates of transmission and reception of data by the

USART's were determined by the programmable timer circuits. The timers were software programmed with the appropriate countdown number and effectively divided the master clock frequency of 1.2288 Megahertz by that countdown number. The outputs of the timer circuits were jumper connected to the Transmit Clock (TxC) and Receive Clock (RxC) pins on the respective USART's.

1. Printer Interface

The Teletype Model 40 Printer interface required the consideration of handshaking signals between the USART's on the SBC 534 board and printer to maximize the speed of transmission while avoiding any data overrun. Connections between the SBC 534 and Model 40 were as indicated in Fig.

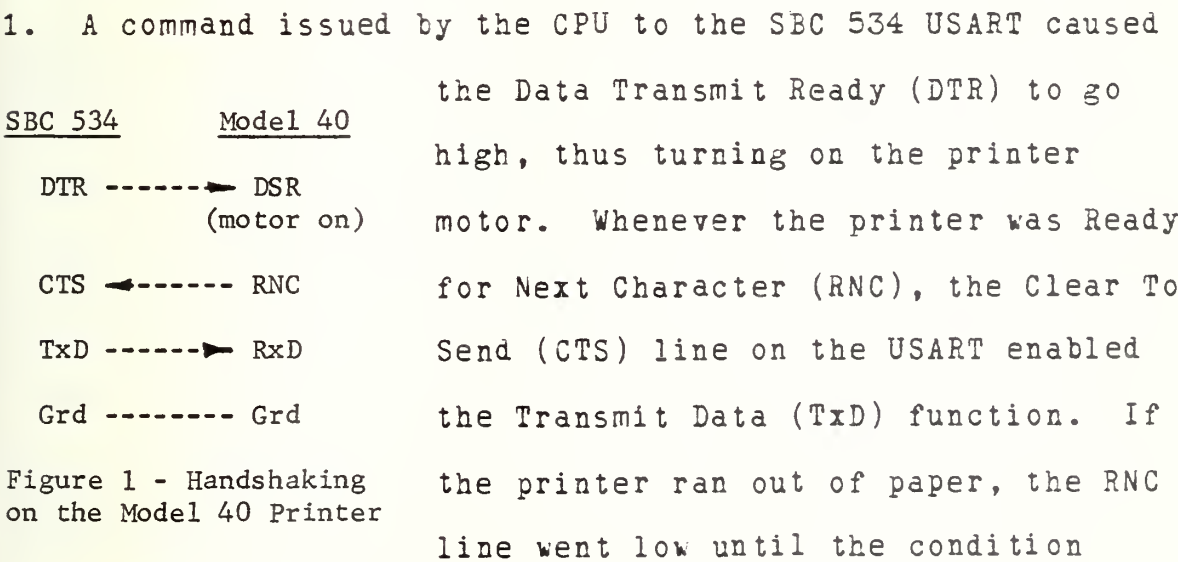


Figure 1 - Handshaking on the Model 40 Printer

was rectified. Since data transmission was one-way from microprocessor to printer, other handshaking facilities were not needed.

Of two one-byte data buffers involved in the transmit function of the USART, one actually transmitted the data words serially (similar in operation to a shift register). This action was enabled by the CTS line indicating that the printer was ready to receive. The second buffer accepted data words from the CPU and loaded the first buffer in parallel at the proper time. The full or empty condition of the second buffer could be determined during program control by checking the value of the Transmitter Empty (TxE) bit in the USART status word.

2. High-speed Line Interface

In the high-speed line interface, there was no handshaking between the SBC 534 USART and the IBM 2701 unit.

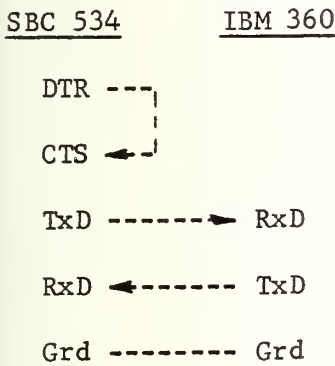


Figure 2 - Handshaking on the high-speed line

The only hardware consideration was how to enable the Clear To Send (CTS) line on the USART. By permanently connecting the Data Transmit Ready (DTR) and CTS lines on the USART, the CTS and thus the transmit data (TxD) function were enabled by setting the DTR bit to high in the command word from the CPU to the USART. The

obvious problems associated with the absence of handshaking were solved through software provisions.

3. Analog to Digital Converter Interface

The ST-800 converter was already configured except for minor changes to accommodate DMA operation. It was installed in the MDS-800 mainframe, and connected to the SBC-501 DMA controller board and the analog input terminal. All handshaking between the ST-800 and DMA controller was automatic as described in Ref. 2.

The scan-clock option, which provided for a selectable delay between scans, was enabled by jumper connection. Since an external scan initiation was desired, pin 34 on the ST-800 J2 connector was grounded. Effectively, the scan-clock option circuitry was used to initiate each scan. The actual signal came not from the scan clock, however, but instead from a negative TTL pulse which was input at pin 36 of the ST-800 J2 connector from an external pulse generator.

4. Direct Memory Access Interface

The Intel Direct Memory Access controller board was installed in the MDS-800 mainframe and connected to the ST-800 converter. The DMA was set to operate at interrupt level four by adjusting a rotary switch on the board. Upon

completion of a cycle, the DMA generated a signal to the CPU interrupt controller which then stopped program execution in order to service the interrupt.

III. SOFTWARE

All programming on the project was done using options available under the CP/M (Control Program/Monitor) monitor. This operating system allows the user to manage files on disk and provides the basic input/output facilities necessary to communicate with peripheral devices. System utilities allow the user to create, edit, load, run, and record programs on the diskette. Two powerful programs, the Macro Assembler (MAC) and the Symbolic Instruction Debugger (SID), give the user vast capabilities to assemble and monitor programs at execution in order to easily detect errors. The system is a product of Digital Research and is described in Ref. 10.

The programming language options available were assembly language and PL/M. Assembly language is shorthand notation for machine language which allows mnemonic instructions, with a one-to-one correspondence between each assembly instruction and a machine code instruction. Because of this, assembly affords direct control over the working registers of the central processor unit; however, for the same reason even simple jobs for the microprocessor can result in long and complex programs. Programs must first be assembled, whereby the assembly mnemonics are compiled into

hex code and addresses are assigned to symbols. Next the program must be loaded, or converted to binary code, before execution by the microprocessor. The only alternative is the PL/M language which is somewhat more sophisticated but which, when reduced finally to binary code, results in about a twenty-five percent waste of memory. The 8080 assembly language was therefore used in all programs during this project.

All assembly programs devised for this project required the use of large memory buffers, so efficiency of programming was paramount in order to reserve as much memory space as possible. For the commonly used 48K system, for example, the memory locations 0-100H and A900H-BFFFH were used for the operating system code. If the user program occupied storage locations 100H-1000H, only 39,078 locations (A900H-1000H) remained available for data storage.

Another observed disadvantage of the assembly language was that the programs were difficult to follow even when well-documented. For this reason, all the programs were designed to be "user oriented" with a multitude of prompts and explanatory comments being echoed to the CRT. Additionally, the programs were heavily documented and instruction guides written for each interface.

The software which interfaced the analog to digital converter, high-speed line, and printer to the micro-processor could all be classified as monitor and control programs. The peripheral devices were monitored and controlled by the central processor unit while keeping the user informed via the CRT.

A. PRINT Program

The PRINT program searches the disk for a specified file, loads the file into memory buffer, and outputs the file to the Model 40 Printer. While outputting the file, PRINT also creates a format for the standard 11x14 inch paper, numbers the pages, and heads each page of printout with the given filename and filetype. If desired by the user, the PRINT program will double space the output; this option works in conjunction with the single/double space switch inside the printer cabinet. PRINT is compatible with all ASCII filetypes.

Another option allows the partial printout of a file between two specified strings of data. This feature is especially useful when working with large files and conserves both paper and time.

Most source files residing on the user's disk are not pre-formatted, hence the PRINT program produces a neat,

orderly output with numbered and titled pages. Certain files, however, including PRN files generated by the Macro Assembler or the Tex Formatter, have already been formatted for a similar output. In order to avoid double formatting, an option exists in the PRINT program whereby the user is queried whether the named file is already formatted. An affirmative response causes the formatting and page numbering features of the program to be suppressed.

1. Printer Control

The program's first task is to initialize the printer and to output data at a rate commensurate with the printer's ability. The CPU first sets up the appropriate timer on the SBC 534 board to pace the binary output at 9600 bits per second. Next the USART is commanded to transmit seven bit words (the eighth bit is zero for all ASCII characters) with one start bit, one stop bit, and no parity bit. The entire serial word train involves ten bits of data. Additionally the CPU command resets any USART error flags and drives the DTR line high, thus turning on the printer motor.

Once the USART is initialized, the CPU reads its status and checks the condition of the Transmitter Empty (TxE) flag. As soon as the transmitter buffer is determined to be empty, the CPU outputs the next data byte.

2. File Reading

Using CP/M system functions, the file to be printed is found and read from the diskette. Since the CP/M disk read function reads 128 byte blocks of data at once, another CP/M function is used to increment the memory location by 128 for each block of data read from the diskette. This process continues until the byte "1AH" is encountered signifying the end of file (EOF).

3. Formatting

Counters are maintained to limit each line to 131 characters and each page to 55 lines. At the beginning of each page the page number, filename, and filetype are output. At the end of each line the keyboard is checked for a user interrupt. The process continues until the end of file (EOF) byte is again encountered. At this time the program turns off the printer motor and returns to the CP/M environment.

4. Prompts

Once the program is executed, user prompts flow sequentially to the CRT and the responses are checked for reasonableness. Any problems associated with incorrect responses, file reading, or control of the printer result in

automatic error messages to the console.

5. PRINT User's Guide

The PRINT User's Guide was intended to be used as an independent manual. The guide provides detailed operating instructions for the Model 40 Printer interface and is included as Appendix E. A listing of the PRINT Assembly program is included as Appendix I.

B. LINK Program

Programming for the high-speed line interface was difficult because the absence of handshaking on the line presented some unique problems. When transmitting from the microprocessor to the IBM 360, the rate and regularity at which data words were output were of no significance. The IBM 2701 unit received one complete line before answering. Upon receiving a byte "13H" (XOFF) signalling the end of a line, the 2701 unit answered with a sequence of bytes: "0DH" (carriage return), "0AH" (line feed), "00H" (null), "3EH" (CMS prompt ">"), and "11H" (XON). Any information transmitted by the IBM 360 always preceded this exact sequence. The programmed arrangement was, therefore, that each unit would take turns transmitting and receiving.

More complicated provisions had to be inserted into the program, however. If the microprocessor attempted to transmit a line containing more than 132 characters, the 2701 unit rejected the excess characters and interrupted with an error message. Also there were occasional instances when the IBM 360 output a large number of lines without the XON. For example, if commanded to print a FORTRAN file, the IBM 360 would output the entire file before transmitting the XON. Therefore, the capability of interrupting the IBM 360 was needed. Instead, the control program had to allow for reception while transmitting and for transmission while receiving.

This was accomplished by setting up two separate loops for the transmit and receive functions. When involved in the reception of characters, the microprocessor CPU constantly checked the keyboard for a user interrupt. If one were found, the program immediately issued a pair of XON characters to the 271 unit while still receiving characters. When the 2701 received the XON's, it acknowledged the interrupt with the usual sequence.

When involved in the transmission of characters, the CPU constantly checked the receive buffer for a data word. When one was found, the program control reverted to the receive function.

1. USART Setup

The USART and timer for the high-speed line were set up similarly to the printer USART. The timer was commanded to generate a baud rate of 1200 baud and the USART was commanded to both transmit and receive. The transmitted serial word train contained one start bit, seven data bits, and two stop bits. The only available baud rate on the high-speed line was 1200 baud. Future improvements to the rate are discussed in the conclusion section to this thesis.

2. Monitor Function

When executed, the LINK program was in the receive status. After receiving the first transmission from the IBM 360, program control went into the transmit function. While in this status, the CPU program alternated between checking the receive buffer for an interrupt and checking the keyboard for a user input. Upon receipt of a user input, the CPU screened the input for certain control characters and, if one were found, branched to the proper subroutine. This monitor function was designed so that control characters used during CP/M operation could also be used when operating with the IBM 360 under CMS. User inputs that were not control characters were output to the IBM 360.

A Control I, the tab command under CP/M, was transmitted to the IBM 360 as a "?" which should have been previously defined to CMS as a logical tab character. A RUBOUT was transmitted as a CMS delete character symbol and a Control U as a delete line symbol. A Control R or Control T caused program control to branch to subprograms that effected the transfer of complete files between micro-processor diskette and IBM disk. Similarly, a Control P caused control to branch to a routine that turned on the printer if off and vice versa. This allowed the user the capability of echoing all correspondence with the IBM 360 to the printer.

If a Control C were input, the program control instituted a soft boot and returned the user to the CP/M environment. The high-speed line was still active although the LINK program was no longer in service. Any transmissions by the IBM 360 at this time "fell on deaf ears". A Control G caused the program to print on the console a list of all Control functions.

3. Data Buffers

Although the high-speed line operating at a baud rate of 1200 baud was usually slower than the microprocessor and all its peripherals, there was one circumstance when the LINK program could not keep pace with the line. If the

printer option were on and a line feed character were being implemented, a delay resulted while waiting for the printer to get ready for the next character. To provide for this circumstance, all data received from the IBM 360 was routed through a First-In-First-Out (FIFO) buffer. After determining that the USART receive buffer did not have a byte ready, the CPU next checked both the CRT and printer to determine if they were ready to receive a byte. If so, the last byte received was output. If either the CRT or printer were not ready, the byte was stored in the FIFO buffer and the USART receive buffer rechecked. In practice the buffer usually expanded after encountering a line feed character because of the printer delay, but caught up before the end of the next line due to the superior baud rates of the CRT (2400) and the printer (9600).

Another type of buffer was utilized in the transmit file and receive file subprograms. A file to be transmitted to the IBM 360 was first completely loaded into memory before transmission, similar to the operation of the PRINT program. If the file size exceeded the available memory, then part of the file was loaded and transmitted, and then another part until the end of the file was encountered. For the 48K system the memory available as a data buffer was about 38K. For files being received from the IBM 360, an insurmountable problem sometimes arose. The file was being received too fast to simultaneously write on the diskette,

so the data had to be buffered. If the file exceeded the available memory, then transmission by the IBM 360 had to be stopped immediately to avoid losing any of the file.

Because of the timesharing operation of the IBM 360 under CMS, the transmission could not be immediately interrupted. Since this anomaly could not be corrected, it was determined that the user would have to limit incoming files to 38K or else break up larger files into 38K segments.

4. LINK User's Guide

Precise instructions for the operation of the LINK program are contained in the LINK User's Guide, Appendix C. The assembly program listing is included as Appendix G.

C. GO Program

The GO program controls the operation of the ST-800 Analog to Digital Converter with the Direct Memory Access Controller. The primary concern in designing this system was to effect the fastest possible data sampling rate while maintaining a high degree of flexibility. The crucial element of speed and the complexity of the component interaction combined to make the software development for this system quite a challenge.

When operating with the DMA, the ST-800 does not communicate directly with the CPU. The DMA is programmed with the total number of converted data bytes to be passed and the memory address at which to store the first byte. The ST-800 is programmed through the DMA with regard to the initial and final channels to be converted. The process of converting the analog signal inputs for the initial through final channels and passing them to the DMA is known as a scan. Full handshaking between the DMA and ST-800 circuits is employed and the throughput time for converting an analog signal into two hex bytes and passing both bytes through the DMA to random access memory is approximately twenty-two microseconds. When one scan is completed, the ST-800 relies on either the CPU or a signal from the scan clock to initiate another scan. When the word length register in the DMA counts down to zero, the DMA has finished its programmed task and waits to be reset.

Initially the approach toward meeting the primary goal was to set up the system on a dual-interrupt basis. Although this scheme provided tremendous flexibility, in some cases it retarded the conversion process from full speed operation. Another configuration was ultimately adopted, but the dual-interrupt approach had some merit and is discussed under the heading of Alternative Solutions.

The Scan-clock Option on the ST-800 provides for initiation of subsequent scans after the first is completed. An end-of-scan signal starts a preset countdown clock which, when timed out, initiates the next scan. The disadvantages to this option were that hardware changes were required to vary the countdown interval, and the fastest scan repetition rate was 1000 scans per second.

By enabling the Scan-clock Option but disabling the countdown timer itself, an external pulse could be applied to initiate scans through the scan-clock circuitry. This method was adopted as the most flexible as well as the fastest.

1. Data File Parameters

The contents of a data file is a collection of hex digits and two such files would be indistinguishable without additional information. The first file of data was named DATA01.XXX and subsequent filenames were incremented by one digit. Through a sequence of user prompts and responses, the program determined which options the user desired. This information was used to set up the data conversion run and also was recorded in the data file to facilitate later identification. Included in the file information block were the initial and final channels, number of data points in the sample, scan repetition rate, run coordination number, and

the number of data bytes involved in each scan.

2. ST-800 and DMA Setup

The number of data points specified by the user was multiplied by two since each digitized data word required two bytes of storage. The result was programmed into the word length register of the DMA. The initial and final channels to be scanned were loaded into the ST-800 via the DMA. The memory location 900H was programmed into the DMA as the future address of the first converted data byte. The DMA controller was then commanded to transfer data from the ST-800 to memory. The ST-800 was commanded by the CPU to start conversion.

3. DMA Reset

Since the pulse generator which initiated subsequent scans was disabled at this point in time, the ST-800 converted through one complete scan and stopped. The word length register on the DMA was not decremented to zero after one scan, hence no interrupt was forthcoming. This first dummy scan was necessary simply to synchronize the ST-800 with the pulse generator.

The word length register and memory address register were now reloaded with their initial values. The DMA was

given a new command word which allowed it complete control of the data bus and the user prompted to enable the pulse generator. By this method the first data byte from the first channel went into the first memory location. The channels were converted at the maximum throughput rate of the ST-800-DMA combination (about 45,000 Hertz) until each scan was completed, and the scan repetition rate coincided with the pulse generator output. When the entire data sample was finished, the word length register decremented to zero and the DMA issued a level four interrupt. A jump vector which had been previously inserted into the RST 04 location directed program control to a routine which serviced the interrupt, disabled the DMA, and prompted the user to disable the pulse generator. Lastly the program wrote the data file to the system diskette if desired by the user and then set up for another run.

4. GO User's Guide

The GO User's Guide, Appendix B, provides the details for setup and operation of the data acquisition system. The GO Assembly program is listed in Appendix F.

D. DATLINK Program

The DATLINK program is a modification of LINK and is identical in most respects. Since the data acquired with

the GO system was recorded on the diskette in hex bytes, each byte had to be converted into two ASCII characters before transmission over the high-speed line. The transmit file mode of DATLINK limited each line to the number of data bytes obtained from each scan. Therefore files created under CMS on the IBM 360 were already formatted with one scan per line.

Because of the additional code needed to accommodate the data files, the receive file mode was removed from the DATLINK program. The User's Guide for DATLINK is included as Appendix D and the Assembly program listing is Appendix H.

IV. SYSTEM QUALIFICATION

System qualification was achieved by digitizing known analog signals, storing the data files on diskette, and transmitting the files to the IBM 360 for data reduction. The output files were then transmitted back to the micro-computer system, stored on diskette, and output to the line printer.

A. Shannon's Sampling Theorem

When digitizing a signal, care must be taken to ensure that Shannon's Sampling Theorem is obeyed; otherwise there is a possibility of aliasing occurring. In general, a degree of conservatism should be followed when digitizing such that ten to fifteen samplings should take place each fundamental period and at least ten to fifteen waveforms should be recorded. If the presence of higher harmonics were suspected, added conservatism should be used.

B. Qualification Test

Sinusoid waveforms with carefully measured frequencies of 20, 200, and 1000 Hertz were chosen for data sampling. The system was set up according to the GO User's Guide,

Appendix B, and the scan triggering pulse generator frequency was measured at 300, 3000, and 10,000 Hertz, respectively. After the data was acquired and stored, the files were sent via the DATLINK program to the IBM 360. Next, using the LINK program, a FORTRAN reduction program was created within the IBM computer similar to the BASIC program reported by Pickelsimer, Ref. 13, and Englehardt, Ref. 1.

C. Data Sampling Theory

One common form of unsteady data recording involves periodic natural signals of arbitrary waveform having a well-established fundamental frequency. As an example, instrumentation transducer system transfer functions would involve data records at various prescribed frequencies of input and output signals. The systems described in this thesis are naturally oriented for providing transfer function type of information using the following cross-correlation scheme to pick out the Fourier components of a deterministic type waveform. Consider a data set $X(1)$, $X(2)$, $X(3)$, ..., $X(N)$ representing a waveform of a known frequency which has been sampled at given intervals. After truncating the set to an integral number of periods, the bias or average value can be determined and removed from each member of the set.

D. Fourier Analysis\

Any periodic waveform can be represented by the Fourier Series

$$X(t) = \sum_{n=1}^{\infty} [A_n \cos n\omega_1 t + B_n \sin n\omega_1 t + A_0]$$

and the coefficients can be found by

$$A_0 = 1/T \int_0^T X(t) dt$$

$$A_n = 2/T \int_0^T X(t) \cos n\omega_1 t dt$$

$$B_n = 2/T \int_0^T X(t) \sin n\omega_1 t dt$$

In cases where the data set represents a known simple waveform (no harmonics) such as the sinusoid used in the system qualification, the Fourier coefficients can be obtained by an estimation procedure. For the assumed truncated data set with bias removed

$$Y(1), Y(2), Y(3), \dots, Y(M)$$

representing a discretized sinusoid signal with frequency F and scan rate of ΔT , the first harmonic estimates become

$$A = (2/M) \sum_{I=1}^M Y(I) \cos [2 \text{ Pi } F \Delta T (I)]$$

$$B = (2/M) \sum_{I=1}^M Y(I) \sin [2 \text{ Pi } F \Delta T (I)]$$

and the magnitude and phase are estimated by

$$C = [A^2 + B^2]^{1/2}$$

$$\phi = \text{Tangent Inverse } [-B/A]$$

Higher harmonics, such as the Kth, can be estimated by replacing $[2 \text{ Pi } F \Delta T (I)]$ with $[2 \text{ Pi } (K) F \Delta T (I)]$ in the above equations.

Had the data set $X(1), \dots, X(N)$ resulted from a random waveform, the above formulae conceptually would be replaced by applying a Fast Fourier Transform algorithm to the data set. This procedure is built into several existing programs in the Computer Center library.

E. Interchannel Sampling Delay

The Fourier Coefficient estimation procedure described above was used during system qualification to establish the interchannel sampling delay. The scan rate or sampling rate refers to the time involved between converting the (Ith) and (Ith + 1) samples of a specific input channel. This scan rate is adjustable since it is controlled by an external pulse generator serving as a trigger. Whenever more than one channel is being digitized, there is a slight time difference between the instants of sampling for the respective channels. This time difference is known as the interchannel sampling delay and is not adjustable since it is established

by the throughput rate of the Analog to Digital converter-DMA controller combination.

F. REDUCE Fourier Coefficient Program

The FORTRAN program created to reduce the system qualification data was similar to the BASIC program used by Englehardt in Ref. 1. Since the test signals were simple waveforms with known frequencies, the estimation procedure described above was used. The REDUCE Fortran Program, listed as Appendix M, was written to accommodate data from four input signals. Since the same test signal was applied to each of the four input channels, the phase differences evident in the reduced data sets gave a close determination of the interchannel sampling delay (21.7 microseconds).

G. System Qualification Results

The reduced data from the three test runs are presented in Appendix N. The sinusoid waveforms had identical magnitudes and that fact was reflected on all four channels of data for each of the three test runs. The magnitudes of the second harmonics were approximately 0.3 percent of the first harmonic magnitude in each case. The existence of a second harmonic was attributable to slight imperfections in the sinusoid generator used for the test waveforms.

The most significant finding from the reduced data was the interchannel sampling delay. For each test run, the difference in phase between two consecutive channels, when divided by the period of the test waveform, indicated a delay of approximately 21.7 microseconds. The throughput rate for the combination of Analog-to-Digital converter and DMA controller was faster than had been predicted. Therefore the maximum sampling rate of the data acquisition system was determined to be slightly in excess of 45,000 Hertz, as compared to the initial value of 40,000 Hertz estimated.

V. ALTERNATIVE SOLUTIONS

The existence of multiple solutions to a specific problem leads to a variety of approaches in microcomputer application. Hardware selection between commercially manufactured or user-constructed devices, the choice of hardware or software to accomplish a given task, and the infinite approaches of software itself exemplify some of the decisions facing the potential user.

Initially a circuit board was constructed for the purpose of driving the Model 40 Printer and high-speed line. Many design problems were encountered and valuable experience was gained. However, the Intel SBC 534 Input/Output Board was later utilized because of its capacity for future system improvement.

A. Dual-Interrupt Data Acquisition

The concept first implemented in setting up the Analog to Digital Converter and the Direct Memory Access controller was to use a timer circuit contained on the SBC 534 board to initiate each scan. A jumper selectable option on the SBC 534 permitted the series operation of two timers. One timer served as a clock for the second timer which initiated an

interrupt signal after counting down to zero. The DMA controller and SBC 534 board were hardwired to generate level four and level five interrupts, respectively. The DMA controller and Analog to Digital Converter were programmed for one complete scan followed by an interrupt. The timers and interrupt controller on the SBC 534 board were programmed to delay for a specific interval before interrupting. Starting both processes together, the program waited for the DMA controller interrupt indicating the end of the scan, and then reset the DMA controller. When the timed interrupt occurred, a software routine reset the timers and re-initiated the two circuits. When the desired number of data points had been converted, the program disabled the interrupt mechanism and wrote the data on the system diskette.

While the operational details of the dual-interrupt setup are contained in the GO2 program listing, Appendix J, this approach was ultimately replaced by the system already described. Two substantial obstacles to its successful operation were never overcome. The presence of the SBC 534 board installed in the MDS mainframe caused a level five interrupt during the bootstrap operation resulting in an aborted disk drive interface. A patch inserted into the CP/M BIOS program averted the untimely interrupts, but a more significant problem remained.

The interrupt service routines were long and cumbersome, particularly the routine that reset the SBC 534 timers. In order to effect the exact desired interval between scans, the time required to implement the reset instructions was taken into account by modifying the countdown interval to a value of 100 microseconds less than the scan interval. This difference was estimated by totalling the instruction cycle times in the routine. Also, the DMA interrupt service routine had to be completed before the timer interrupt occurred so as to avoid stacked interrupts. As shown in Fig. 3, the allowable conversion time of approximately twenty-two microseconds per channel (1 - 2) was 150 microseconds less than the scan period.

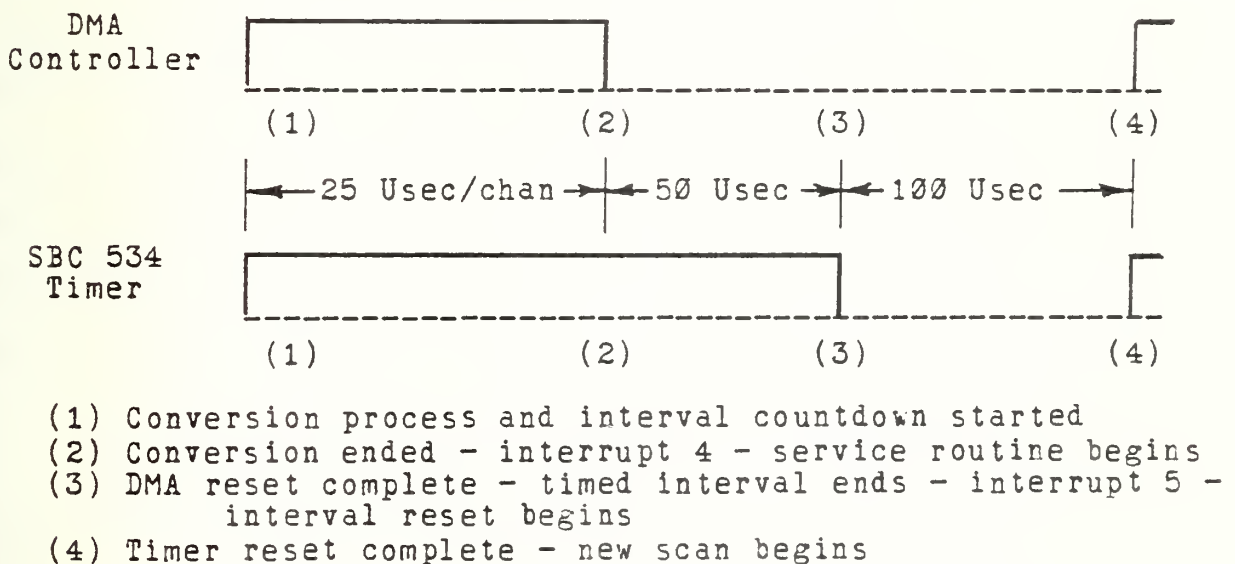


Figure 3 - Dual Interrupt Timing

These software delays resulted in a considerable limitation on the maximum scan rates. With the dual-interrupt process,

the eight channel scan rate was only 2800 Hertz and the one channel rate was 5000 Hertz. With the externally timed system described earlier, the eight channel and one channel scan rates were 5000 Hertz and 45,000 Hertz, respectively.

B. Model 40 Printer as a List Device

The CP/M system provides for the operation of a "list" device which originally was designated as the teletype terminal. Several routines within the BIOS program and the MDS monitor divert the microprocessor output to the list device. For example, the CP/M routines TYPE and PIP, as well as the monitor function LO (for List Out), are directed to the list facility. Additionally, by depressing a Control P key, the user can cause all characters directed to the console to also be echoed to the list device. Before the printer can be used as the CP/M list device, it must be initialized by a separate routine such as the ON Assembly program which is included as Appendix L, and the CP/M itself must be altered to address the printer.

A simple patch to the CP/M BIOS program, included as Appendix K, can be used to alter the system so that output to the list device can be redirected to the Model 40 Printer. If the printer USART were programmed beforehand to accept data, the patched CP/M could produce a printed copy of all the information presented on the console. The patch

may be implemented under DDT control and the patched file can be used to generate a patched system disk.

VI. CONCLUSIONS

The data acquisition system developed during this project provided an extremely flexible, dynamic tool for investigating rapidly changing experimental aerodynamic phenomena. Signals from analog measuring devices were sampled at a maximum rate of 45,000 times per second and the data stored on magnetic disks. The data was then expeditiously transferred to the IBM 360 computer where higher level language programs directed the efficient reduction of raw data to formatted answers. The empirical results were then returned to local microprocessor environment and printed. The printer was operated alone to produce hard copy source listings, records of microprocessor functions, and text formatted printouts such as this document.

A. Future System Improvements

The speed at which data files were transmitted to the IBM 360 computer was limited by the IBM 2701 Data Adapter unit to 1200 baud or about 120 characters per second. Although the rate increase over earlier interfaces was by a factor of eleven to one, the capability exists to further improve the speed another eight times to a rate of 9600 bits per second. The MDS system including hardware and software

was designed to run at the higher speed and only minimal software changes would be necessary to effect such an improvement. Because other users cannot accommodate the 9600 baud, the IBM 2701 unit is hardwired to operate at only 1200 baud.

The scheduled expansion of the IBM interface for high speed line operation will provide a line hardwired to operate at 4800 baud. Whenever the IBM facilities are modified, the microprocessor can be upgraded by making some minor changes to the LINK and DATLINK programs. The countdown number applied to the high-speed line USART should be altered in both programs to generate the faster baud rate. Also, during operation under the receive file mode of the LINK program, a subroutine "CONOUT" echoes all received characters to the CRT terminal. Since the CRT baud rate of 2400 baud is less than 4800, the instruction "CALL CONOUT" (08BEH) should be deleted.

APPENDIX A

Glossary

ASCII: American Standard Code for Information Interchange. This is a seven-bit-plus-parity code established by the American National Standards Institute to achieve compatibility between data services.

assembler: a compiler that translates assembly language into hex code and assigns memory locations to labels.

assembly language: programming language used in microcomputer applications.

baud: a serial data transmission rate expressed in bits per second.

BIOS: Basic Input/Output Operating System - a subprogram of the CP/M system that effects all transfers of information between the CPU and its peripheral devices.

bit: binary digit - a single unit of information in a binary word.

buffer: a block of random access memory that has been reserved for temporary data storage.

byte: an eight-bit binary word which is processed as a single quantity.

CMS: Cambridge Monitoring System - a time sharing scheme used by the IBM 360 computer which allows several users simultaneous access to a single virtual machine.

CRT: cathode ray tube - a television-like picture tube used in visual display terminals.

CP/M: Control Program/Monitor - a software system which allows the microprocessor to be operated as a microcomputer. The system is described in Ref. 10.

CPU: Central Processor Unit - the area of the microcomputer

that computes and controls all logical and arithmetic functions.

DMA: Direct Memory Access - a facility whereby input/output data can be transferred to/from memory without passing through the CPU.

FIFO: First-In-First-Out - a buffer in which data is inserted and removed in the same order.

hardware: the physical circuitry and related devices within the microprocessor.

Hertz: units of rate of repetition (cycles per second).

hex: number system based on 16 decimal - one hex digit equates to four binary bits; e.g., 14 decimal is E hex or 1110 binary.

instruction cycle: a finite time span during which the CPU executes programmed instructions. For the MDS this time span can be as short as 2 microseconds. The instruction cycle time may be computed by multiplying the number of clock cycles in a given instruction by 0.5 microseconds.

interrupt: an independent circuit and logic system within the microcomputer. Certain peripheral devices can signal the interrupt logic controller which screens interrupt priorities so that several simultaneous signals can be processed. The interrupt controller halts program execution and diverts the CPU's attention to a subroutine that services the interrupt.

K: symbol used to denote one kilo-byte (1024 decimal or 400 hex bytes) of memory.

machine code: the bit patterns actually used by the CPU to execute its assigned logic functions.

MDS: Microcomputer Development System - the Central Processor Unit with related memory and peripheral devices.

peripheral device: any major independent component controlled by the CPU; e.g., the CRT, teletype, printer, disk drive, or Analog to Digital Converter.

PL/M: Programming Language/Medium.

RAM: random access memory - volatile memory area used for program code and data storage.

RS-232C driver: a transistorized switching device which converts TTL voltage levels to +/- 15 volts for longer range transmission. The RS-232C refers to an Electronic Industries Association (EIA) specification for the device.

ROM: Read Only Memory - non-volatile memory in a computer which contains permanent machine code.

software: the program which contains routines to operate the microcomputer.

throughput: refers to the elapsed time for one complete cycle; e.g., the Analog to Digital Converter throughput includes the time to sample and convert an input, pass the digitized word to the DMA, and set up for the next cycle.

TTL: Transistor Transistor Logic - low current logic devices operate with five volts D. C. power supplies. Subsequently a logical true state is indicated by +5 volts and a false state by 0 volts.

Usec: microsecond - one millionth of a second.

USART: Universal Synchronous Asynchronous Receiver Transmitter - integrated circuit device which converts parallel transmissions into serial transmissions and vice versa.

XON: an ASCII "11" which signifies the beginning of a transmission.

XOFF: an ASCII "13" which signifies the end of a transmission.

APPENDIX B

GO USER'S GUIDE

I. CAPABILITIES

A. GO INTERFACES THE INTEL MDS 800 MICROPROCESSOR AND DIRECT MEMORY ACCESS CONTROLLER BOARD WITH THE DATEL ST-800 ANALOG TO DIGITAL CONVERTER BOARD FOR HIGH SPEED DATA ACQUISITION. A MAXIMUM OF 16 CHANNELS OF ANALOG DATA CAN BE INPUT, CONVERTED, AND STORED IN RANDOM ACCESS MEMORY AT A RATE OF 45 KHZ.

B. GO INTERFACES A SEQUENCE OF PROMPTS AND USER RESPONSES. THESE RESPONSES ARE USED BY THE PROGRAM TO SET UP THE ANALOG TO DIGITAL CONVERTER AND DIRECT MEMORY ACCESS CONTROLLER TO PROVIDE A LEVEL FOUR INTERRUPT WHEN DATA HAS BEEN ACQUIRED.

C. GO WRITES EACH BLOCK OF ACQUIRED DATA ONTO A FLOPPY DISK FOR LATER RETRIEVAL. EACH DATA FILE CONTAINS FORMATTED PARAMETERS WHICH DESCRIBE THE DATA SAMPLING PROCEDURES, SUCH AS NUMBER OF DATA POINTS, SCAN RATE, AND A RUN COORDINATION NUMBER WHICH IS ENTERED BY THE USER.

D. A VARIABLE FREQUENCY PULSE GENERATOR IS USED DURING THE DATA ACQUISITION PROCESS TO INITIATE EACH SCAN. CARE MUST BE TAKEN TO AVOID SELECTING A SCAN RATE WHICH EXCEEDS THE SYSTEMS CAPABILITY. FIGURING A THROUGHPUT TIME OF TWENTY-TWO MICROSECONDS PER CHANNEL FOR CONVERSION TO MEMORY STORAGE, THE SELECTED PULSE RATE SHOULD NOT EXCEED 45,000 DIVIDED BY THE NUMBER OF CHANNELS; E.G., IF EIGHT CHANNELS WERE TO BE SAMPLED, THE SCAN RATE SHOULD NOT EXCEED 5500 SCANS PER SECOND.

E. SUCCESSIVE DATA SAMPLING RUNS ARE RECORDED ON THE FLOPPY DISK IN DRIVE B WITH FILENAMES DATA01.XXX, DATA02.XXX, ETC. IF A LIKE FILENAME ALREADY EXISTS ON THE DISK, IT IS DELETED BEFORE THE NEW FILE IS WRITTEN.

II. SETUP

A. ANALOG INPUTS ARE LIMITED TO PLUS OR MINUS FIVE VOLTS AND SHOULD BE CONNECTED TO THE SYSTEM THROUGH A LOCALLY CONSTRUCTED INPUT TERMINAL. THE ANALOG TO DIGITAL CONVERTER CAN THEN BE CALIBRATED BY EXECUTING A DATEL TEST PROGRAM ST-800 (AVAILABLE ON DISK AND PAPER

TAPE IN THE MICROPROCESSOR LAB).

B. A NEGATIVE TTL PULSE (WHICH STROBES ZERO VOLTS) IS ALSO CONNECTED TO THE INPUT TERMINAL. A DIGITAL FREQUENCY COUNTER SHOULD BE INTERCONNECTED TO OBTAIN PRECISE SCAN RATE INFORMATION. THE PULSE GENERATOR SHOULD BE TESTED AND THEN PLACED IN A STANDBY CONDITION (NO PULSING).

C. A PREFERABLY BLANK, FORMATTED DISKETTE SHOULD BE PLACED IN DISK DRIVE B.

III. OPERATION

THE GO PROGRAM IS EXECUTED BY THE FOLLOWING COMMAND:

GO <CARRIAGE RETURN>

IMMEDIATELY THE USER IS PROMPTED WITH

ENTER STARTING CHANNEL

FOLLOWING USER'S REPLY, THE NEXT PROMPT APPEARS:

ENTER FINAL CHANNEL

NOTE: RESPONSE TO THE ABOVE TWO PROMPTS SHOULD BE IN THE RANGE OF 0 - 15. IF THIS RANGE IS EXCEEDED OR IF THE STARTING CHANNEL IS GREATER THAN THE FINAL CHANNEL, ANOTHER PROMPT APPEARS:

TRY AGAIN, TURKEY

AND THE ABOVE PROMPTS ARE REPEATED.

NEXT THE USER IS PROMPTED WITH A CHOICE OF DATA BLOCK SIZES:

ENTER DESIRED NUMBER OF DATA POINTS

ENTER	DATA POINTS	DISK SPACE
A	1024	2K
B	4096	8K
C	10240	20K
D	20480	40K
E	26624	52K(62K SYSTEM)

THE USER SELECTS ONE OF THE OPTIONS BY TYPING THE APPROPRIATE LETTER AND A CARRIAGE RETURN.

USER IS THEN PROMPTED WITH

ENTER SCAN RATE

THIS RESPONSE CAN BE ENTERED IN ANY FORMAT

NOTE: THE ACTUAL SCAN RATE IS DETERMINED BY THE PULSE GENERATOR. THE RESPONSE TO THE ABOVE PROMPT WILL APPEAR IN THE FILE INFORMATION PARAMETERS.

THE NEXT PROMPT IS

ENTER COORDINATION NUMBER

THIS RESPONSE CAN BE ANYTHING THE USER MIGHT CHOOSE TO DISCRIMINATE BETWEEN VARIOUS RUNS.

FINALLY THE SYSTEM INDICATES A READY CONDITION BY

START PULSE GENERATOR

AT THIS TIME OR WHENEVER USER CHOOSES, THE PULSE GENERATOR SHOULD BE CHANGED FROM A STANDBY TO PULSING CONDITION. THE COMPLETION OF A RUN IS SIGNALLED BY A BEEP AND

RUN COMPLETE - DISABLE PULSE

THE PULSE GENERATOR SHOULD BE RETURNED TO A STANDBY CONDITION AT THIS TIME. THE USER IS PROMPTED WITH

WRITE DATA FILE ON DISK?? (Y/N)

IF USER SELECTS ANY KEY BUT "N", THE PROGRAM WILL ECHO THE FILE PARAMETERS TO THE CONSOLE FOR USER VERIFICATION AND WRITE THE DATA FILE ONTO THE DISKETTE IN DRIVE B. ANY PROBLEM INCURRED IN THE WRITE PROCESS WILL BE DETAILED BY EITHER

DISK WRITE ERROR - TRY ANOTHER

OR

DISK FULL

AFTER PLACING A CLEAN DISK IN DRIVE B, USER SHOULD TYPE A CARRIAGE RETURN TO START THE WRITE PROCESS AGAIN.

NOTE: REGARDLESS WHETHER THE DATA ACQUIRED IN A RUN IS WRITTEN ON A DISK, THE DATA FILENAME WILL BE INCREMENTED.

THE NEXT PROMPT TO APPEAR IS

ANOTHER DATA RUN DESIRED?? (Y/N)

SELECTION OF Y WILL START THE PROMPTS AGAIN, AND SELECTION OF ANY OTHER KEY WILL REBOOT THE SYSTEM AND

RETURN USER TO CPM.

NOTE: IF THE PROGRAM IS NOW RE-EXECUTED, THE DATA
FILENAME COUNT WILL START OVER AT DATA01.XXX AND
OVERWRITE PREVIOUS DATA FILES.

IV. DATA FILES

AN ACQUIRED DATA FILE CAN BE DUMPED UNDER CP/M. THE
FIRST 128 BYTE BLOCK OF THE FILE CONTAINS INFORMATION
RELATING TO ITS ACQUISITION. A SAMPLE DUMPED FILE
FOLLOWS:

```
44 41 54 41 30 31 01 07 31 30 32 34 24 35 30 30
30 24 30 30 39 31 31 30 30 33 24 00 00 00 00 00
12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 ETC
```

THE FIRST SIX BYTES ARE THE FILENAME IN ASCII
DATA01

THE NEXT TWO BYTES ARE THE INITIAL AND FINAL CHANNELS
IN HEX

01,07

THE NEXT THREE PARAMETERS ARE ASCII CODE INDICATING THE
NUMBER OF DATA POINTS, SCAN RATE, AND RUN COORDINATION
NUMBER, EACH FOLLOWED BY THE DELIMITER "\$"

1024

5000

00911003

THE '12' INDICATES THAT 1200H WAS THE UPPER LIMIT ON
MEMORY USED -

THE '10' IS THE HEX REPRESENTATION OF THE NUMBER OF
MEMORY BYTES PER SCAN

THE REMAINDER OF THE BLOCK IS ZEROES

M. T. ELLIOTT, NPGS
AUGUST 28, 1978

APPENDIX C

LINK USERS GUIDE

I. LINK INTERFACES THE MDS 800 (AND MODEL 40 PRINTER) WITH CP/CMS THROUGH A 1200 BAUD TELEPHONE LINE. BOTH THE LINE AND THE PRINTER ARE DRIVEN BY 8251 USARTS INCORPORATED IN AN SBC534 I/O BOARD. LINK OPERATES IN ONE OF THREE MODES AS FOLLOWS:

A. DIRECT LINKUP MODE

1. TRANSMITS CHARACTERS TYPED ON KEYBOARD TO CP/CMS WITH SOME FILTERING BUT NO BUFFERING; ECHOES CHARACTERS TO CONSOLE (AND PRINTER)
2. RECEIVES CHARACTERS FROM CP/CMS AND UTILIZES A FIFO BUFFER TO PRINT THE CHARACTERS ON THE CONSOLE (AND PRINTER)
3. ALTHOUGH NO HANDSHAKING IS UTILIZED ON THE LINE, SOFTWARE PROVISIONS ALLOW EITHER END TO INTERRUPT THE OTHER'S TRANSMISSIONS
4. CERTAIN CHARACTERS TYPED ON THE KEYBOARD ARE FILTERED OUT:

RUBOUT - BACKSPACES THE CONSOLE AND TRANSMITS A DELETE CHARACTER SYMBOL "@"

CONTROL I - ECHOES AND TRANSMITS A "?" TO INDICATE A LOGICAL TAB - NOTE: "?" MUST BE PREVIOUSLY DEFINED TO THE CMS AS A TAB CHARACTER

CONTROL U - TRANSMITS A DELETE LINE SYMBOL "["

CARRIAGE
RETURN - TRANSMITS END OF LINE SYMBOL AND WAITS FOR AN ANSWER

ADDITIONAL CONTROL CHARACTERS ALTER PROGRAM EXECUTION AS FOLLOWS:

CONTROL C - REBOOTS SYSTEM

CONTROL D - RETURNS USER TO DIRECT LINKUP
MODE

CONTROL P - TURNS PRINTER ON IF OFF AND VICE
VERSA

CONTROL R - INITIALIZES "RECEIVE FILE" MODE

CONTROL T - INITIALIZES "TRANSMIT FILE" MODE

B. TRANSMIT FILE MODE

1. AUTOMATICALLY ISSUES ALL CP/CMS COMMANDS TO
EFFECT THE TRANSFER OF AN ENTIRE FILE FROM FLOPPY
DISK TO CP/CMS P-DISK

2. LINEFEED CHARACTERS APPEARING IN THE FLOPPY DISK
FILES ARE FILTERED OUT; HOWEVER, TAB CHARACTERS ARE
CONVERTED TO "?" AND TRANSMITTED TO CP/CMS

3. THE PRINTER DOES NOT WORK IN THIS MODE

NOTE: WHEN TRANSMITTING CONTINUOUS DATA FILES, THE
PROGRAM SETS THE LINE LENGTH AT 132 CHARAC-
TERS (83H). THE NAMED CMS FILETYPE MUST
ACCOMMODATE THIS RECORD LENGTH. IF A SHORT-
ER LINE LENGTH IS DESIRED, THE PROGRAM CAN BE
ALTERED UNDER DDT AT PROGRAM COUNT OF 984H.

C. RECEIVE FILE MODE

1. AUTOMATICALLY ISSUES ALL CP/CMS COMMANDS TO
EFFECT THE TRANSFER OF AN ENTIRE P-DISK FILE TO THE
FLOPPY DISK

2. THE DATA BEING RECEIVED IS ECHOED TO THE CONSOLE
FOR THE CONVENIENCE OF THE USER

3. THE TRANSMISSION BY CP/CMS CAN BE INTERRUPTED BY
DEPRESSING ANY KEY. THIS ACTION RESTORES USER TO
THE "DIRECT LINKUP" MODE AND THE CMS IS SHIFTED INTO
CP. THE TERMINATED FILE IS LOST ALTHOUGH THE FILE-
NAME WILL EXIST IN THE DISK DIRECTORY.

II. OPERATION

A. DIRECT LINKUP MODE

THE PROGRAM IS EXECUTED AS FOLLOWS:

LINK <CR>

THE USER IS PROMPTED WITH

DIAL 2721 FOR LINE -- TYPE CARRIAGE RETURN
A CONNECTED LINE IS INDICATED BY THE MESSAGE

CP-67 ON LINE

NORMAL LOGIN PROCEDURE AND CP/CMS TYPING CONVENTIONS
ARE USED AND ANY KEY WILL "BREAK" THE CMS TRANSMIS-
SIONS

B. TRANSMIT FILE MODE

UPON INITIALIZATION BY CONTROL T, USER IS PROMPTED
WITH

DISK:FILENAME.FILETYPE

THE FILE TO BE TRANSMITTED SHOULD BE ENTERED EXACTLY
ACCORDING TO THIS FORMAT. IF FORMAT IS VIOLATED,
THE USER IS PROMPTED WITH

REPEAT

IF THE NAMED FILE CANNOT BE FOUND AS LISTED, THE
APPROPRIATE PROMPT APPEARS

FILE NOT FOUND

AND USER IS RETURNED TO THE "DIRECT LINKUP" MODE.
ASSUMING PROPER ENTRY OF THE FILE TO BE TRANSMITTED,
THE NEXT PROMPT IS

CMS FILENAME FILETYPE?

THE FORMAT OF THE ANSWER TO THIS PROMPT IS NOT SPEC-
IFIED BUT NOTE THAT THE CMS FILENAME WILL BE EXACTLY
AS ENTERED.

NOTE: IF A KNOWN MISTAKE IS MADE IN ANSWERING THE
ABOVE PROMPTS, TYPING CONTROL U WILL ALLOW
USER TO START THE LINE AGAIN.

NOTE: THE CMS FILENAME SHOULD BE A NEW FILE SO THE
CMS WILL SHIFT DIRECTLY INTO "INPUT" MODE.

AFTER ENTERING THE FILENAMES, THE PROGRAM OPERATES
AUTOMATICALLY BUT ECHOES ITS COMMANDS TO CMS ON THE
CONSOLE SO THE USER IS AWARE OF THE PROGRAM STATUS

NOTE: TYPING CONTROL D WILL IMMEDIATELY RETURN USER
TO THE "DIRECT LINKUP" MODE

----- SAMPLE TRANSMITTED FILE -----

```
LINK:  DISK:FILENAME.FILETYPE
USER:  A:LINK.ASM<CR>
LINK:  CMS FILENAME FILETYPE?
USER:  HOOKER FORTRAN<CR>
LINK:  EDIT HOOKER FORTRAN
CMS:   >EDIT HOOKER FORTRAN
      >NEW FILE
      >INPUT:
LINK:  >TRANSMITTING
CMS:   >EDIT
LINK:  >SAVE
CMS:   >INPUT:
LINK:  >RELOADING
      TRANSMITTING
CMS:   >EDIT
LINK:  >FILE
CMS:   >R;
LINK:  >TRANSMISSION COMPLETE
      0034 RECORDS TRANSMITTED
      >
```

THE USER IS AUTOMATICALLY RETURNED TO THE "DIRECT LINKUP" MODE AT THIS TIME.

NOTE: IF THE FILE TO BE TRANSMITTED EXCEEDS THE BUFFER OF 40K BYTES, THE PROGRAM COMMANDS CMS TO SAVE THAT PORTION OF THE FILE, THEN 40K MORE BYTES ARE READ AND TRANSMITTED.

NOTE: FLOPPY DISK RECORDS ARE 128 BYTES IN LENGTH; P-DISK RECORDS ARE 829 BYTES IN LENGTH. DEPENDING ON THE CMS FILETYPE USED, ONE CMS RECORD EQUALS FROM ONE TO FOUR MDS RECORDS.

C. RECEIVE FILE MODE

UPON INITIALIZATION BY CONTROL R, THE FOLLOWING PROMPT APPEARS:

CMS FILENAME FILETYPE?

FORMAT REQUIREMENTS ARE SIMILAR TO THOSE ABOVE FOR "TRANSMIT FILE MODE". THE NEXT PROMPT IS

DISK:FILENAME.FILETYPE

AND AGAIN THE FORMAT IS THE SAME.

NOTE: THE FLOPPY DISK FILENAME AND FILETYPE SHOULD BE NEW TO THE DISK. THE PROGRAM WILL DELETE ANY EXISTING FILE WITH THE SPECIFIED FILENAME AND FILETYPE!!!!

IF DISK SPACE IS LIMITED, ONE OF THESE PROMPTS WILL APPEAR:

NO DIRECTORY SPACE AVAILABLE

(APPEARS BEFORE FILE IS TRANSMITTED BY CMS)

OR

DISK FULL

(APPEARS AFTER FILE HAS BEEN TRANSMITTED AND INDICATES FILE LENGTH EXCEEDED THE AVAILABLE DISK SPACE)

IN BOTH CASES, USER IS RETURNED TO THE "DIRECT LINKUP" MODE.

NOTE: TYPING CONTROL D WILL IMMEDIATELY RETURN USER TO THE "DIRECT LINKUP" MODE

ASSUMING NO DISK PROBLEMS, THE PROGRAM OPERATES AUTOMATICALLY.

----- SAMPLE RECEIVED FILE -----

```
LINK:  CMS FILENAME FILETYPE?
USER:  FOURPLAY OUTPUT72<CR>
LINK:  DISK:FILENAME.FILETYPE
USER:  HOWCUM.HEX<CR>
LINK:  PRINT FOURPLAY OUTPUT72
      RECEIVING
CMS:   :54424A2031303948534B37363231304D5F
      :ETC ETC ETC
      :ETC ETC
      :ETC
      >R;
LINK:  >TRANSMISSION COMPLETE
      0078 RECORDS TRANSMITTED
      >
```

THE USER IS AUTOMATICALLY RETURNED TO THE "DIRECT LINKUP" MODE.

NOTE: IF THE FILE TO BE RECEIVED FROM CMS EXCEEDS THE BUFFER SIZE OF 40K BYTES, THE REMAINDER OF THE FILE WILL BE LOST.

NOTE: IF USER ELECTS TO TERMINATE FILE RECEPTION, DEPRESSING ANY KEY WILL RETURN PROGRAM CONTROL TO "DIRECT LINKUP" AND THE CMS WILL BE INTERRUPTED

A HANDY REFERENCE GOUGE FOR "LINK" FOLLOWS:

LINK

<CR>	END OF LINE
RUBOUT	DELETE CHARACTER
CONTROL C	REBOOT
CONTROL D	RETURN TO DIRECT LINKUP
CONTROL I	TAB "?"
CONTROL P	PRINTER ON/OFF
CONTROL R	RECEIVE FILE MODE
CONTROL T	TRANSMIT FILE MODE
CONTROL U	DELETE LINE
BREAK	ANY KEY INTERRUPTS

MACK T. ELLIOTT, NPGS
AUGUST 22, 1978

APPENDIX D

DATLINK USERS GUIDE

I. DATLINK IS A MODIFICATION OF THE LINK PROGRAM DESIGNED SPECIFICALLY FOR TRANSFERRING DATA FILES FROM FLOPPY DISK TO CP/CMS P-DISK.

A. DIRECT LINKUP MODE - THIS MODE OPERATES EXACTLY THE SAME AS IN THE LINK PROGRAM

B. TRANSMIT FILE MODE

1. DATA FILES ACQUIRED AND WRITTEN ON THE FLOPPY DISK BY THE GO PROGRAM ARE IN HEX CODE. THE FIRST FILE RECORD (128 BYTES) CONTAINS THE DATA FILENAME, INITIAL AND FINAL CHANNELS OF EACH SCAN, THE SCAN RATE, NUMBER OF DATA POINTS IN THE RUN, AND RUN COORDINATION NUMBER. ADDITIONALLY, THE FIRST FILE RECORD CONTAINS THE MOST SIGNIFICANT BYTE OF THE UPPER LIMIT ON MEMORY SPACE USED, AND THE NUMBER OF MEMORY BYTES USED PER SCAN (NUMBER OF CHANNELS TIMES TWO).

2. THE TRANSMIT FILE MODE ECHOES THE FILE PARAMETERS TO THE CONSOLE AND IMMEDIATELY BEGINS TRANSMISSION OF THE FILE TO CP/CMS. EACH HEX BYTE OF DATA IS CONVERTED TO TWO ASCII CHARACTERS BEFORE TRANSMISSION. THE LINE LENGTH IS SET AT THE NUMBER OF BYTES PER SCAN TO FACILITATE LATER FORMATTING FOR USE IN IBM 360 PROGRAMMING. E.G., THE MAXIMUM LINE LENGTH THAT COULD OCCUR WOULD BE 64 CHARACTERS (16 CHANNELS TIMES TWO BYTES PER CHANNEL TIMES TWO ASCII CHARACTERS PER BYTE).

3. THE MAXIMUM SIZED DATA FILE THAT CAN BE TRANSMITTED IS 40K (52K WITH A 62K SYSTEM) CORRESPONDING TO THE LARGEST DATA SAMPLE THAT CAN BE ACQUIRED WITH THE GO PROGRAM. ALSO, THE NUMBER OF FILE RECORDS TRANSMITTED IS NOT COUNTED AND DISPLAYED WITH THE DATLINK PROGRAM.

C. THE RECEIVE FILE MODE DOES NOT EXIST IN THE DATLINK PROGRAM.

II. OPERATION

A. DIRECT LINKUP MODE - EXECUTION OF THE DATLINK PROGRAM AND OPERATION OF THE "DIRECT LINKUP" MODE IS EXACTLY THE SAME AS FOR THE LINK PROGRAM.

B. TRANSMIT FILE MODE

UPON INITIALIZATION BY CONTROL T, THE PROMPTS AND REPLIES ARE THE SAME AS FOR THE LINK PROGRAM. BEFORE TRANSMISSION BEGINS, THE USER IS PROMPTED WITH THE DATA FILE PARAMETERS.

----- SAMPLE TRANSMITTED FILE -----

```
DATLINK: DISK:FILENAME.FILETYPE
USER:    B:DATA03.XXX
DATLINK: CMS FILENAME FILETYPE
USER:    FILE FT01F001
DATLINK: DATA03
          1024 DATA POINTS
          5000 SCANS PER SECOND
          RUN COORDINATION NUMBER 822001
          EDIT FILE FT01F001
CMS:     >EDIT FILE FT01F001
          >NEW FILE
          >DEFAULT PARAMETERS SET
          >INPUT
DATLINK: >TRANSMITTING
CMS:     >EDIT
DATLINK: >FILE
CMS:     >R;
DATLINK: >TRANSMISSION COMPLETE
          >
```

C. RECEIVE FILE MODE - UPON INITIALIZATION BY CONTROL R, THE USER IS PROMPTED WITH

TO RECEIVE FILE, USE LINK PROGRAM

THE MESSAGE IS SELF-EXPLANATORY

NOTE: ALL PROMPT REPLY FORMATS, ERROR MESSAGES, AND CONTROL CHARACTER USAGE IS EXACTLY THE SAME AS IN THE LINK PROGRAM.

M. T. ELLIOTT, NPGS
AUGUST 22, 1978

APPENDIX E

PRINT USER'S GUIDE

I. CAPABILITIES

A. PRINT INTERFACES THE INTEL MDS 800 WITH THE TELETYPE MODEL 40 HIGH SPEED PRINTER THROUGH AN INTEL SBC 534 INPUT/OUTPUT BOARD. PRINT ACCESSES FILES STORED ON FLOPPY DISK AND TRANSMITS THEM TO THE PRINTER AT A 9600 BAUD RATE.

B. FOR DISK FILES ALREADY FORMATTED, SUCH AS PRN FILES GENERATED BY THE TEX FORMATTER OR THE MACRO ASSEMBLER (PRODUCTS OF DIGITAL RESEARCH), THE PRINT PROGRAM OUTPUTS THE FILE WORD FOR WORD TO THE PRINTER.

C. ALL OTHER FILES STORED ON FLOPPY DISK IN ASCII CODE ARE FORMATTED BY PRINT FOR THE STANDARD 11 X 14 PAPER USED IN THE PRINTER. PRINT PROVIDES FOR ONE INCH MARGINS ON THE BOTTOM AND BOTH SIDES AND A THREE QUARTER INCH MARGIN AT THE TOP. EACH PAGE OF THE PRINTED FILE IS HEADED BY THE FILENAME, FILETYPE, AND PAGE NUMBER. PRINTED FILES ARE NORMALLY SINGLE SPACED, BUT A DOUBLE SPACE OPTION MAY BE SELECTED AND SHOULD COINCIDE WITH THE SPACING SWITCH ON THE PRINTER.

D. FOR PARTIAL PRINTOUTS OF LARGE FILES, THE USER CAN ENTER TWO STRINGS OF UP TO FIFTEEN CHARACTERS EACH, AND THE PROGRAM WILL SEARCH THE FILE AND PRINT ONLY THE TEXT BETWEEN THE STRINGS.

E. THE PRINT PROGRAM LOADS THE ENTIRE FILE INTO RANDOM MEMORY BEFORE COMMENCING OUTPUT TO THE PRINTER. IF THE AVAILABLE MEMORY (40K BYTES) IS EXCEEDED BY THE NAMED FILE, THEN 40K BYTES ARE PRINTED AND THEN ANOTHER 40K BYTES ARE LOADED AND PRINTED.

F. THE PRINT OPERATION CAN BE INTERRUPTED AT ANY TIME BY THE USER.

II. OPERATION

THE PRINT PROGRAM IS EXECUTED BY THE FOLLOWING COMMAND:

PRINT <DISK:>FILENAME.FILETYPE

THE PROGRAM TURNS ON THE PRINTER MOTOR AND SEARCHES FOR THE NAMED FILE. IF THE FILE CANNOT BE OPENED AS LISTED, THE FOLLOWING PROMPT APPEARS:

FILE NOT FOUND

DONE

AND THE USER MUST RE-EXECUTE USING THE CORRECT DISK/FILENAME/FILETYPE. AFTER THE FILE IS OPENED, USER IS PROMPTED WITH

TEXT FILE?? (Y/N)

IF THE FILE HAS BEEN GENERATED BY THE TEX FORMATTER OR THE MACRO ASSEMBLER, NO FURTHER FORMATTING BY THE PRINT PROGRAM IS NEEDED. THE USER SHOULD TYPE YES (Y) AND THE FILE WILL PRINT AS FORMATTED. IF NO (N) IS SELECTED, THE NEXT PROMPT IS

TYPE 2 FOR DOUBLE SPACE
(DEFAULT = SINGLE SPACE)

TYPING ANY KEY OTHER THAN "2" WILL RESULT IN SINGLE SPACING.

NOTE: SELECTION OF DOUBLE SPACING MUST COINCIDE WITH THE SPACING SWITCH SETTING ON THE PRINTER.

NEXT THE USER IS PROMPTED WITH

PRINT ALL (A) OR PART (P)??

IF ANY KEY OTHER THAN "P" IS SELECTED, THE PROGRAM WILL PRINT THE ENTIRE FILE. IF "P" IS SELECTED, ANOTHER PROMPT APPEARS:

ENTER STRING1,STRING2

EITHER STRING MAY BE OMITTED, BUT THE COMMA MUST BE INCLUDED.

NOTE: THE PRINTOUT WILL INCLUDE THE FIRST STRING AND EXCLUDE THE SECOND STRING.

AT ANY TIME THE MODEL 40 IS PRINTING, USER MAY INTERRUPT BY TYPING ANY KEY. THE FOLLOWING PROMPT WILL APPEAR:

TYPE K TO CANCEL OR SPACE TO CONTINUE

THIS MESSAGE IS SELF-EXPLANATORY.

AFTER COMPLETING THE PRINTOUT, THE PRINTER IS TURNED

OFF BY THE PROGRAM. THE FOLLOWING MESSAGE APPEARS ON THE CONSOLE:

DONE

A SOFT BOOT BY THE PROGRAM RESTORES USER TO CPM.

NOTE: IF THE PRINTER POWER SWITCH IS OFF OR THE PRINTER RUNS OUT OF PAPER, THE PRINT PROGRAM IDLES UNTIL THE CONDITION IS RECTIFIED, THEN RESUMES PRINTING.

M. T. ELLIOTT, NPGS
AUGUST 25, 1978

70


```

STKBTM EQU $ ;INITIATE STACK POINTER HERE
;
;
;MESSAGES
;
;
MSG1: DB CR,LF,LF,'ENTER STARTING CHANNEL $'
MSG2: DB CR,LF,LF,'ENTER FINAL CHANNEL $'
MSG3: DB CR,LF,LF,'START PULSE GENERATOR ----',CR,LF,LF,'$'
MSG4: DB CR,LF,LF,'TRY AGAIN, TURKEY $'
M45: DB 'DATA POINTS$'
MSG5: DB CR,LF,LF,'ENTER DESIRED NUMBER OF DATA POINTS '
DB CR,LF,LF,'ENTER DATA POINTS DISK SPACE'
DB CR,LF,LF,'A
M5A: DB '1024 2K',CR,LF
DB B
M5B: DB '4096 8K',CR,LF
DB C
M5C: DB '10240 20K',CR,LF
DB D
M5D: DB '20480 40K',CR,LF
DB E
M5E: DB '26624 52K (62K SYSTEM)',CR,LF,'$'
MSG6: DB CR,LF,LF,'ENTER
M6A: DB 'SCAN RATE $'
MSG65: DB CR,LF,LF,'ENTER
M65A: DB 'COORDINATION NUMBER $'
MSG7: DB CR,LF,LF,'WRITE DATA FILE ON DISK?? (Y/N) $'
MSG8: DB CR,LF,LF,'ANOTHER DATA RUN DESIRED?? (Y/N) $'
MSG9: DB CR,LF,LF,'DISK FULL - TRY ANOTHER - RETURN WHEN READY $'
MSG10: DB CR,LF,LF,'DISK WRITE ERROR - TRY ANOTHER - RETURN WHEN READY$'
MSG11: DB CR,LF,LF,'RUN COMPLETE - DISABLE PULSE',CR,LF,LF,'$'
;
;
;
;

```


0360	314501	LXI	SP,	STKBTM	;SET UP STACK POINTER
0363	3EC3	MVI	A,	JUMP	;JUMP INSTRUCTION
0365	322000	STA	R04		;SET UP INTERRUPT
0368	218104	LXI	H,	RESET4	;ADDR OF INT 4 ROUTINE
036B	222100	SHLD	R04+1		
;CHANGE CPU MASK TO ACCEPT RST 04 INTERRUPTS					
036E	3E6E	MVI	A,	6EH	;ALLOWS RST 0,4,7
0370	D3FC	OUT	MASK		
;SPECIFY DISK DRIVE B FOR ALL DATA WRITES					
0372	0E0E	MVI	C,	14	
0374	1E01	MVI	E,	1	;DRIVE B
0376	CD0500	CALL	BDOS		
;GET VALUES FOR INITIAL AND FINAL CHANNELS AND WORD LENGTH					
;SETUP:					
0379	CD4E25	CALL	RECORD		;ZERO OUT RECORD LINE
037C	CD9404	CALL	DIGIT1		;GETS CHANNEL VALUES
037F	216508	LXI	H,	MEMORY+6	;INITIAL CHANNEL VALUE
0382	3A8608	LDA	MEMORY+7		;FINAL CHANNEL VALUE
0385	96	SUB	M		;DETERMINE DIFFERENCE
0386	F28F03	JP	DIFF		
0389	CDFC04	CALL	OOPS		;FINAL CAN'T BE LESS
038C	C37903	JMP	SETUP		;BACKUP AND TRY AGAIN
;DIFF:					
038F	C601	ADI	1H		;NUMBER WORDS PER
0391	17	RAL			;SCAN IS NUMBER OF
0392	32B008	STA	MEMORY+30H		;CHANS TIMES TWO
;DATPT:					

03D7 013902	DPOINT:	LXI	B,	M5D	
03DA 3EA0		MVI	A,	0A1H	
03DC C3E403		JMP	DOWN		
03DF 014802	EPOINT:	LXI	B,	M5E	
03E2 3ED8		MVI	A,	0D9H	
03E4 C5	DOWN:	PUSH	B		
03E5 322401		STA	COUNT		;MSB OF WORD LENGTH
03E8 C609		ADI	9H		
03EA 32A008		STA	MEMORY+20H		;FOR THE RECORD
03ED 118708		LXI	D,	MEMORY+7	
03F0 CDF404		CALL	KEY		
03F3 C1		POP	B		
03F4 FE0D		CPI	CR		;WAIT FOR CARR RETURN
03F6 CAFF03		JZ	DLOOP		
03F9 CDFC04		CALL	OOPS		
03FC C39503		JMP	DATPT		
	; DLOOP:				
03FF 0A		LDAX	B		COPY NUMBER DATA POINTS INTO RECORD
0400 FE09		CPI	09H		
0402 CA0B04		JZ	DLEND		;LOOK FOR TAB CHAR
0405 12		STAX	D		
0406 03		INX	B		
0407 13		INX	D		
0408 C3FF03		JMP	DLOOP		
040B 3E24	DLEND:	MVI	A,	'\$'	
040D 12		STAX	D		
040E 13		INX	D		
	; ;GET PARAMETERS AND SAVE FOR THE RECORD				
	; ;				


```

0441 3E24      MVI    A,'$'
0443 12        STAX   D
; BEGIN:
0444 CD5E04    CALL    DMASET          ;SETS UP DMA AND ST800
0447 D342     OUT     DMA+2H         ;RUNS ONE SCAN
; NOW READY TO BEGIN SCANNING WHEN PROMPTED
0449 117701    LXI     D,MSG3
044C 0E09      MVI     C,9H
044E CD0500    CALL    BDOS
; RESET DMA WORD LENGTH REG AND MEMORY ADDR REG,
; CHANGE COMMAND WORD TO GIVE DMA COMPLETE
; CONTROL OF THE SYSTEM BUS
;
0451 D349      OUT     DMA+9H
0453 CD6E04    CALL    SYNC
0456 3E37      MVI     A,DMACMD+00100000B ;RESETS DMA FOR RUN
0458 D34A      OUT     DMA+0AH
; DATA ACQUISITION STARTS WITH PULSE GENERATOR
; NOTHING TO DO BUT WAIT
; WAIT:
045A AF        XRA     A
045B C35A04    JMP     WAIT
; END OF MAIN PROGRAM
;
;
;
;
;
;
;

```



```

; SYNC:
046E AF      XRA      A      ;LSB OF LENGTH REG
046F D34C    OUT      DMA+0CH
0471 3A2401  LDA      COUNT  ;MSB OF LENGTH REG
0474 D34D    OUT      DMA+0DH
0476 210009  LXI      H,      MEMORY+80H
0479 7D      MOV      A,      L      ;LSB OF MEMORY ADDR
047A D34E    OUT      DMA+0EH
047C 7C      MOV      A,      H      ;MSB OF MEMORY ADDR
047D D34F    OUT      DMA+0FH
047F FB      EI
0480 C9      RET

;
; DMA NOW READY TO GO WHEN COMMAND WORD IS ISSUED
;
; RESET4:
0481 D349    OUT      DMA+9H      ;RESET DMA
0483 3E20    MVI      A,      REVRT ;CLEARS INT 4 FROM CPU
0485 D3FD    OUT      0FDH      ;INTERRUPT PENDING STACK
0487 F1      POP      PSW      ;KEEP STACK STRAIGHT
0488 113D03  LXI      D,      MSG11 ;GET USER TO TURN OFF
048B 0E09    MVI      C,      9H  ;PULSE GENERATOR
048D CD0500  CALL     BDOS
0490 FB      EI
0491 C30705  JMP      DONE      ;REENABLES INTERRUPTS
                                ;GO PROCESS DATA

;
; ROUTINE TO READ IN INITIAL AND FINAL CHANNELS
;
; DIGIT1:
0494 114501  LXI      D,      MSG1      ;PROMPT USER
0497 0E09    MVI      C,      9      ;GET ENTERED CHARACTER
0499 CD0500  CALL     BDOS
049C CDF404  CALL     KEY
049F FE0D    CPI      CR
04A1 CA9404  JZ       DIGIT1

```


04A4 D630	SUI	30H	;REDUCE ASCII
04A6 328508	STA	MEMORY+6	
04A9 CDF404	CALL	KEY	
04AC FE0D	CPI	CR	;SEE IF SECOND CHAR
04AE CAC604	JZ	DIGIT2	
04B1 D630	SUI	30H	;REDUCE ASCII
04B3 C60A	ADI	0AH	;CONVERT TO HEX
04B5 328508	STA	MEMORY+6	
04B8 CDF404	CALL	KEY	
04BB FE0D	CPI	CR	;STILL NEED CR
04BD CAC604	JZ	DIGIT2	
04C0 CDFC04	CALL	OOPS	;TOO MANY CHARACTERS
04C3 C39404	JMP	DIGIT1	;TRY AGAIN

04C6 116001	LXI	D,	MSG2
04C9 0E09	MVI	C,	9
04CB CD0500	CALL	BDOS	
04CE CDF404	CALL	KEY	;PROMPT USER
04D1 FE0D	CPI	CR	;GET CHARACTER
04D3 CAC604	JZ	DIGIT2	
04D6 D630	SUI	30H	;CR NOT ALLOWED YET
04D8 328608	STA	MEMORY+7	
04DB CDF404	CALL	KEY	
04DE FE0D	CPI	CR	;GET NEXT CHAR
04E0 C8	RZ		
04E1 D630	SUI	30H	;FINISHED IF CR
04E3 C60A	ADI	0AH	;CONVERT TO HEX
04E5 328608	STA	MEMORY+7	
04E8 CDF404	CALL	KEY	
04EB FE0D	CPI	CR	
04ED C8	RZ		
04EE CDFC04	CALL	OOPS	;FINISHED IF CR
04F1 C3C604	JMP	DIGIT2	;TOO MANY CHARACTERS

```

;
;
; DIGIT2:

```

```

;
```



```

;
;
; ROUTINE TO RETRIEVE CHARACTER FROM KEYBOARD
;
KEY:
04F4 D5          PUSH      D
04F5 0E01        MVI       C, 1H
04F7 CD0500      CALL      BDOS
04FA D1          POP       D
04FB C9          RET

;
; ROUTINE PRINTS MESSAGE IF TOO MANY CHARACTERS
;
OOPS:
04FC D5          PUSH      D
04FD 119601      LXI       D, MSG4
0500 0E09        MVI       C, 9
0502 CD0500      CALL      BDOS
0505 D1          POP       D
0506 C9          RET

;
; DONE:
0507 119202      LXI       D, MSG7
050A 0E09        MVI       C, 9H
050C CD0500      CALL      BDOS
050F CDF404      CALL      KEY, 'N
0512 FE4E        CPI
0514 CA2005      JZ        GETMOR
0517 CD0606      CALL      CRLF
051A CD0606      CALL      CRLF
051D C36C05      JMP       FLFILE

;
; GETMOR:
;SEE IF USER WANTS
;FILE WRITTEN
;CHECK ANSWER
;IF NO, CONTINUE
;IF YES, GO WRITE

```


0520 11B502	LXI D,	MSG8	SEE IF USER WANTS
0523 0E09	MVI C,	9H	;ANOTHER RUN
0525 CD0500	CALL BDOS		
0528 CDF404	CALL KEY		;CHECK ANSWER
052B FE59	CPI 'Y'		;IF YES, GO BACK
052D CA3305	JZ RERUN		
			;OTHERWISE, ITS TIME TO QUIT
			;EXIT:
0530 C30000	JMP 0H		;WARM BOOT
			;SET UP FOR ANOTHER RUN
			;RERUN:
0533 3A0901	LDA FLNAME+6		;INCREMENT FILE NAME
0536 3C	INR A		
0537 320901	STA FLNAME+6		
053A FE3A	CPI 3AH		
053C C27903	JNZ SETUP		
053F D60A	SUI 0AH		
0541 320901	STA FLNAME+6		
0544 3A0801	LDA FLNAME+5		
0547 3C	INR A		
0548 320801	STA FLNAME+5		
054B C37903	JMP SETUP		
			;RECORD:
054E 3E00	MVI A,	0H	
0550 118008	LXI D,	MEMORY	
0553 0680	MVI B,	80H	
			;ZERO OUT FILE
0555 12	STAX D		;RECORD WHICH WILL
0556 13	INX D		;CONTAIN PROCESS
0557 05	DCR B		;INFORMATION
0558 C25505	JNZ RDLOOP		


```

055B 010401      ;FILL IN FILENAME          FLNAME+1
055E 118008      LXI B,                      MEMORY
0561 2605      LXI D,                      6H
                                MVI H,
RLOOP2:
0563 0A      LDAX B
0564 12      STAX D
0565 03      INX B
0566 13      INX D
0567 25      DCR H
0568 C26305      JNZ RLOOP2
056B C9      RET

;
;NEXT ROUTINE CREATES AND WRITES A DISK FILE -
;THE FIRST FILE RECORD CONTAINS INFORMATION
;WHICH WILL FACILITATE LATER RETRIEVAL OF THE
;DATA ---
;THE FIRST FILE RECORD CONTAINS THE DATA FILE
;NAME, FIRST CHANNEL, FINAL CHANNEL, NUMBER
;OF DATA POINTS, SCAN RATE, AND RUN CONTROL
;NUMBER - ALSO THE NUMBER OF WORDS PER SCAN
;THE REMAINDER OF THE FIRST FILE RECORD IS ZEROES
;
;
;FLFILE:
;
;CREATE FILE ON DISK DRIVE B
;
056C 0E13      MVI C,          19
056E 110301      LXI D,          FLNAME
0571 CD0500      CALL BDOS
                                ;DELETE OLD FILE, SAME NAME
;CLEAN UP FILE CONTROL BLOCK
0574 AF      XRA A
0575 320F01      STA FLNAME+12
0578 321001      STA FLNAME+13

```


057B 321101	STA	FLNAME+14	
057E 321201	STA	FLNAME+15	
	;CREATE NEW FILE		
0581 0E16	MVI	C,	22
0583 110301	LXI	D,	FLNAME
0586 CD0500	CALL	BDOS	
0589 FEFF	CPI	255	
058B CA2706	JZ	NOROOM	
058E AF	XRA	A	
058F 322301	STA	FLNAME+32	
	;CREATE NEW FILE		
	;RETURNS 255 IF NOT		
	;ENOUGH DISK SPACE		
	;ZERO IT		
	;NEXT RECORD COUNT		
	;WHILE DISK WRITE OCCURS, ECHO DATA FILE PARAMETERS		
	;TO CONSOLE FOR CORRELATION		
	;		
0592 11AB01	LXI	D,	M45
0595 0E09	MVI	C,	9H
0597 CD0500	CALL	BDOS	
059A CD0606	CALL	CRLF	
059D 118708	LXI	D,	MEMORY+7
05A0 CD1706	CALL	CONSL	
05A3 CD0606	CALL	CRLF	
05A6 D5	PUSH	D	
05A7 116A02	LXI	D,	M6A
05AA 0E09	MVI	C,	9H
05AC CD0500	CALL	BDOS	
05AF D1	POP	D	
05B0 CD0606	CALL	CRLF	
05B3 CD1706	CALL	CONSL	
05B6 CD0606	CALL	CRLF	
05B9 D5	PUSH	D	
05BA 117D02	LXI	D,	M65A
05BD 0E09	MVI	C,	9H
05BF CD0500	CALL	BDOS	
05C2 D1	POP	D	
05C3 CD0606	CALL	CRLF	


```

05C6 CD1706      CALL    CONSL
05C9 CD0606      CALL    CRLF

;
;
;
;SINCE DMA PUT PAIRS OF DATA BYTES INTO MEMORY IN REVERSE
;ORDER, WANT TO REVERSE THEM BEFORE WRITING ON DISK
;
FLIP:
05CC 3AA008      LDA     MEMORY+20H          ;UPPER LIMIT ON MEMORY
05CF 210009      LXI     H,    MEMORY+80H    ;BEGINNING OF DATA

;
FLOP:
05D2 46          MOV     B,    M            ;GET LSB
05D3 23          INX     H
05D4 4E          MOV     C,    M            ;GET MSB
05D5 70          MOV     M,    B            ;PUT LSB
05D6 2B          DCX     H
05D7 71          MOV     M,    C            ;PUT MSB
05D8 23          INX     H
05D9 23          INX     H
05DA BC          CMP     H,    H            ;CHECK AGAINST LIMIT
05DB C2D205      JNZ     FLOP

;
;DATA PAIRS NOW IN CORRECT ORDER
;
;
;READY TO START WRITING ONTO DISK
;
FWRITE:
05DE 118008      LXI     D,    MEMORY ;INFO RECORD

;
FLOOP:
05E1 D5          PUSH    D                ;SAVE POINTER
05E2 0E1A        MVI     C,    26         ;

```


05E4	CD0500	CALL	BDOS				
05E7	110301	LXI	D,	FLNAME			;CHANGE BUFFER ADDRESS
05EA	0E15	MVI	C,	21			
05EC	CD0500	CALL	BDOS				;WRITE ONE RECORD
05EF	D1	POP	D				;RETRIEVE POINTER
05F0	F5	PUSH	PSW				;WILL CHECK LATER
05F1	218000	LXI	H,	80H			
05F4	19	DAD	D				;INCREMENT POINTER
05F5	EB	XCHG					;BY 80H
05F6	F1	POP	PSW				
05F7	FE00	CPI	0H				;CHECK FOR WRITE ERRORS
05F9	C23506	JNZ	ERROR				
05FC	3AA008	LDA	MEMORY+20H				;CHECK END OF DATA
05FF	BA	CMF	D				;MSB ONLY
0600	CA4806	JZ	CLOSE				
0603	C3E105	JMP	FLOOP				;GO DO ANOTHER RECORD

0606	D5	PUSH	D				
0607	1E0D	MVI	E,	CR			
0609	0E02	MVI	C,	2H			
060B	CD0500	CALL	BDOS				
060E	1E0A	MVI	E,	LF			
0610	0E02	MVI	C,	2H			
0612	CD0500	CALL	BDOS				
0615	D1	POP	D				
0616	C9	RET					

0617	1A	LDAX	D				
0618	13	INX	D				
0619	FE24	CPI	'\$'				


```

0061B C8      RZ      D      A      2H
0061C D5      PUSH
0061D 5F      MOV      E,
0061E 0E02    MVI      C,
00620 CD0500  CALL     BDOS
00623 D1      POP      D
00624 C31706  JMP      CONSL
;
; ROUTINE INFORMS USER THAT DISK OR DIRECTORY IS FULL
;
;
;
; NOROOM:
00627 11D902  LXI      D,      MSG9
0062A 0E09    MVI      C,      9H
0062C CD0500  CALL     BDOS
0062F CDF404  CALL     KEY
00632 C36C05  JMP      FLFILE
;
;
;
; ERROR:
00635 FE02    CPI      2
00637 CA2706  JZ       NOROOM
0063A 110803  LXI      D,      MSG10
0063D 0E09    MVI      C,      9H
0063F CD0500  CALL     BDOS
00642 CDF404  CALL     KEY
00645 C36C05  JMP      FLFILE
;
; IF ERROR OCCURRED IN WRITING ON DISK, ANOTHER WRITE SHOULD
; BE ATTEMPTED ON ANOTHER DISK
;
;
;
; WHENEVER DATA WRITE IS COMPLETED, NEED TO CLOSE FILE
;
; CLOSE:

```


0648 110301
064B 0E10
064D CD0500
0650 C32005

;
;
;
;
;
;
;

LXI D, FLNAME
MVI C, 16
CALL BDOS
JMP GETMOR
;CHECK WITH USER

END 100H

0653

APPENDIX G

LINK ASSEMBLY PROGRAM

```

0100      ;
0100      EQU C30D04
BDOS      EQU EQU
XON       EQU EQU
XOFF      EQU EQU
CR        EQU EQU
LF        EQU EQU
FF        EQU EQU
EOF       EQU EQU
RUB       EQU EQU
CNTLC     EQU EQU
CNTLD     EQU EQU
CNTLG     EQU EQU
CNTLI     EQU EQU
CNTLP     EQU EQU
CNTLR     EQU EQU
CNTLT     EQU EQU
CNTLU     EQU EQU
FLIMIT    EQU EQU
BUFFMAX   EQU EQU
COUNT:   DS
FCOUNT:   DS
PPREG     DS

5H
11H      ; END OF LINE FROM VIRTUAL MACHINE
13H      ; END OF LINE TO VIRTUAL MACHINE
0DH      ; CARRIAGE RETURN
0AH      ; LINE FEED
0CH      ; FORM FEED
1AH      ; END OF FILE CHAR FOR DISK WRITE
7FH      ; DELETE CHARACTER
03H      ; WARM BOOT "DIRECT LINKUP" MODE
04H      ; RESTORES "DIRECT LINKUP" MODE
07H      ; PRINT INSTRUCTIONS
09H      ; TAB CHARACTER
10H      ; CONTROL P TURNS PRINTER ON AND OFF
12H      ; CONTROL R FOR RECEIVE FILE
14H      ; CONTROL T FOR TRANSMIT FILE
15H      ; DELETE LINE
230H     ; ALLOWS 304 RECORDS OF 128 BYTES
0D000H   ; MAX SIZE OF TRANSFERRED FILE
2        ; COUNT OF RECORDS TRANSFERRED
2        ; FILE COUNT RECORD
1        ; PRINTER CONTROL REG;0 OFF,1 ON

```




;INITIALIZES SBC 534 BOARD

CALL BOARD
IN 60H
;
;TRANSMIT MODE
;

041B CDAE05
041E DB60

TX:

0420 DB61
0422 E602
0424 C2A904
0427 DBF7
0429 E602
042B CA2004
042E 0E01
0430 CD0500
0433 FE0D
0435 CA9204
0438 FE10
043A CA0305
043D FE12
043F CA3D06
0442 FE14
0444 CA0B06
0447 FE03
0449 CA0000
044C FE07
044E CAE604
0451 FE09
0453 CC8C04
0456 FE7F
0458 CC7C04
045B FE15
045D CA8404

0460 4F
0461 FE11
0463 CA7204

IN 61H
ANI 2
JNZ CRCV1
IN 0F7H
ANI 2
JZ TX
MVI C,
CALL BDOS
CPI CR
JZ RCV
CPI CNTLP
JZ PRTCONT
CPI CNTLR
JZ FILERX
CPI CNTLT
JZ FILETX
CPI CNTLC
JZ 00H
CPI CNTLG
JZ GOUGE
CPI CNTLI
CZ CHNG4
CPI RUB
CZ CHNG2
CPI CNTLU
JZ CHNG3

MOV C,
CPI XON
JZ CTX

;CHECKS LINE FOR MESSAGE

;CHECKS KEYBOARD

;LOOPS UNTIL ONE OF THE ABOVE

;READ CHAR FROM CONSOLE
;CHECK FOR CR
;SWITCH TO RECEIVE MODE

;TURN PRINTER ON/OFF

;RECEIVE FILE MODE

;TRANSMIT FILE MODE

;ESCAPE BY REBOOTING
;PRINT INSTRUCTIONS

;TRANSMIT TAB CHAR "?"

;TRANSMIT DELETE CHAR SYMBOL "@"
;TRANSMIT DELETE LINE SYMBOL "["
;AND XOFF

0466 3A0701	LDA	PPREG			
0469 FE00	CPI	Ø			;CHECK IF PRINTER ON
046B CA7204	JZ	CTX			
046E 79	MOV	A,	C		
046F CD2C05	CALL	DRIVER			
CTX:					
0472 79	MOV	A,	C		
0473 CD6F05	CALL	SEND			;SENDS CHAR TO VIRTUAL MACHINE
0476 C32004	JMP	TX			;LOOPS FOREVER
CHNG1:					
0479 3E3F	MVI	A,	'?		
047B C9	RET				
CHNG2:					
047C 3E08	MVI	A,	08H		;BACKSPACE
047E CD3705	CALL	CONOUT			
0481 3E40	MVI	A,	'@'		
0483 C9	RET				
CHNG3:					
0484 3E5B	MVI	A,	'['		
0486 CD6F05	CALL	SEND			
0489 C39204	JMP	RCV			
CHNG4:					
048C 3E3F	MVI	A,	'?		
048E CD3705	CALL	CONOUT			
0491 C9	RET				
;RECEIVE MODE					
RCV:					
0492 3A0701	LDA	PPREG			;CHECK IF PRINTER ON
0495 FE00	CPI	Ø			
0497 CAA404	JZ	CRCV			
049A 3E0D	MVI	A,	CR		;START NEW LINE ON PRINTER
049C CD2C05	CALL	DRIVER			
049F 3E0A	MVI	A,	LF		
04A1 CD2C05	CALL	DRIVER			

04A4 3E13	MVI A,	XOFF	;END OF LINE CHAR
04A6 CD6F05	CALL SEND		
04A9 215A0A	CRCV1: ;HL REGISTER POINTS TO ADDR FOR NEXT WORD RECEIVED		
04AC 115A0A	LXI H,	BUFF	;DE REGISTER POINTS TO ADDR OF NEXT WORD TO BE PRINTED
	LXI D,	BUFF	;FIFO BUFFER ADDR
04AF CD7A05	CALL BREAK		
04B2 DB61	IN 61H		;CHECK LINE FOR CHAR
04B4 E602	ANI 02H		
04B6 CA4205	JZ CKPRT		;IF LINE NOT READY, CHECK IF
			;BUFFER CAUGHT UP
04B9 DB60	IN 60H		;INPUT WORD FROM LINE
04BB E67F	ANI 7FH		
04BD FE11	CPI XON		
04BF CACC04	JZ CATCH		;END OF LINE - LET BUFFER
			;CATCH UP
04C2 FE13	CPI XOFF		
04C4 CAAF04	JZ RX1		;FILTER OUT XOFF CHAR
04C7 77	MOV M,	A	;STORE CHAR
04C8 23	INX H		
04C9 C3AF04	JMP RX1		;LOOP UNTIL END OF LINE
04CC 77	MOV M,	A	;STORE LAST WORD
04CD 1A	LDAX D		;NEXT WORD TO BE PRINTED
04CE FE11	CPI XON		
04D0 CA2004	JZ TX		;GO BACK TO TRANSMIT MODE
04D3 CD3705	CALL CONOUT		;PRINT ON CONSOLE
04D6 3A0701	LDA PPREG		;CHECK IF PRINTER ON
04D9 FE00	CPI 0		

04DB CAE204	JZ	BACK	
04DE 1A	LDAX	D	
04DF CD2C05	CALL	DRIVER	
04E2 13	INX	D	
04E3 C3CD04	JMP	LOOP	
04E6 113D01	LXI	D, MSG2	
04E9 1A	LDAX	D	
04EA FE24	CPI	'5'	
04EC CA2004	JZ	TX	
04EF CD3705	CALL	CONOUT	
04F2 47	MOV	B, A	
04F3 3A0701	LDA	PPREG	
04F6 FE00	CPI	0	
04F8 CAFF04	JZ	GLP	
04FB 78	MOV	A, B	
04FC CD2C05	CALL	DRIVER	
04FF 13	INX	D	
0500 C3E904	JMP	GLOOP	
0503 3A0701	LDA	PPREG	
0506 FE00	CPI	0	
0508 C22005	JNZ	PRTOFF	
050B CDE505	CALL	USART2	
050E 3E01	MVI	A, 1	
0510 320701	STA	PPREG	
0513 3E0D	MVI	A, CR	
0515 CD2C05	CALL	DRIVER	
0518 3E0A	MVI	A, LF	
051A CD2C05	CALL	DRIVER	
051D C32004	JMP	TX	
0520 3E30	MVI	A, 30H	

BACK: ;LOOP UNTIL CAUGHT UP
 GOUGE:
 GLOOP:
 GLP:
 PRTCONT: ;CHECK IF PRINTER ON OR OFF
 ;IF ON, WANT TO TURN OFF
 ;LATER ROUTINES CHECK THIS ADDR
 ;START PRINTER ON NEW LINE
 ;RETURN TO TRANSMIT MODE
 ;CONTROL WORD - TURN PRINTER OFF

0522 D363	OUT 63H		
0524 3E00	MVI A, 0		
0526 320701	STA PPREG		;LATER ROUTINES CHECK THIS ADDR
0529 C32004	JMP TX		
;ROUTINE TO DRIVE PRINTER USART			
0520 F5	PUSH PSW		
	SLO:		
052F 0F	IN 63H		;WAIT UNTIL XMITTER READY
0530 D22D05	RRC		
0533 F1	JNC SLO		
0534 D362	POP PSW		
0536 C9	OUT 62H		
	RET		
;ROUTINE TO DRIVE CONSOLE USART			
0537 F5	CONOUT:		
	SLO2:		
0538 DBF7	PUSH PSW		
053A 0F	IN 0F7H		
053B D23805	RRC		
053E F1	JNC SLO2		
053F D3F6	POP PSW		
0541 C9	OUT 0F6H		
	RET		
;KEEPS TRACK OF WHICH RECEIVED DATA HAS BEEN PRINTED			
0542 7D	CKPRT:		
0543 BE	MOV A, L		
0544 CAAF04	CMP E		
0547 DBF7	JZ RX1		;CAUGHT UP, NO NEED TO PROCEED
0549 0F	IN 0F7H		
054A D2AF04	RRC		
054D 3A0701	JNC RX1		;CONSOLE NOT READY - NO NEED
0550 FE00	LDA PPREG		;TO PROCEED
	CPI 0		;CHECK IF PRINTER ON


```
;CONTROL - DRIVES XMIT LINE LOW
;HOLD LINE LOW FOR 2 WORDLENGTHS
;WAIT 10 MILLISECS
```

3FH
400H

A. 61H B.

```

MVI
OUT
LXI

```

0586 3E3F
0588 D361
058A 010004

DLA1:

DCX	B
MOV	A,
CPI	0
JZ	DLA3
IN	61H
ANI	2
JZ	DLA1
IN	60H
MOV	M,
INX	H

058D 0B
058E 78
058F FE00
0591 CA9F05
0594 DB61
0596 E602
0598 CA8D05
059B DB60
059D 77
059E 23

```

;CHECK LINE FOR CHAR

```

B. 5A0H

LXI

059F 01A005

```
;DELAY 16 MILLISEC
```

DCX	B	B
MOV	A,	
CPI	Ø	
JNZ	DLA2	

05A2 0B
05A3 78
05A4 FE00
05A6 C2A205

RESET:

MV I
 A, 61H
 OUT
 RET
 37H

05A9 3E37
05AB D361
05AD C9

BOARD:

THIS ROUTINE INITIALIZES THE 534 BOARD, THE TIMERS, AND THE TWO USARTS
NEEDED TO DRIVE THE IBM HIGH SPEED LINE AND THE MODEL 40 PRINTER

BASE ADDR OF 534 BOARD	60H
CMMD ADDR OF LINE USRT	61H
DATA ADDR OF LINE USRT	60H
CMMD ADDR OF PTR USRT	63H


```

;
;
;SET UP BOTH USARTS WITH RESETS AND MODE WORDS
;
USART:
04D9 FE00      CPI      0
05D8 3ECA      MVI      A, 0CAH      ;2 STOP, PAR DISABLED, 7 BITS
05DA D361      OUT      61H
05DC 3E5A      MVI      A, 5AH      ;1 STOP, PAR DISABLED, 7 BITS
05DE D363      OUT      63H
05E0 3E37      MVI      A, 37H
05E2 D361      OUT      61H
05E4 C9        RET

USART2:
05E5 3E33      MVI      A, 33H
05E7 D363      OUT      63H
05E9 C9        RET

;THIS SECTION PERTAINS TO TRANSFERRING COMPLETE
;FILES BETWEEN MDS AND IBM 360
;
FCB      EQU      5CH
FCBCN    EQU      FCB+0
FCBFN    EQU      FCB+1
FCBFT    EQU      FCB+9
FCBRL    EQU      FCB+12
FCBRC    EQU      FCB+15
FCB2:    DS       33
FCBCR    EQU      FCB+32

;SUBR PROMPTS CONSOLE FOR FILE TO BE XMITTED, SETS UP FILE
;CONTROL BLOCK, OPENS NEW CMS FILE, TRANSMITS FILE, AND
;RETURNS USER TO DIRECT CMS LINKUP
;
```


FILETX:

060B	3E00	MVI	A,	0	
060D	320301	STA	COUNT		
0610	320401	STA	COUNT+1		
0613	CD6906	CALL	RESTR		;SETS UP FILE CONTROL BLOCK
0616	CD7D07	CALL	CRLF		
0619	CD1C07	CALL	CPNAME		;CP/CMS FILENAME, FILETYPE
061C	CD7D07	CALL	CRLF		
061F	CD8C07	CALL	OPEN		;OPENS DISK FILE
0622	CD8207	CALL	FILERD		;READS DISK FILE
0625	CD4F09	CALL	CMS		;PREPARES CMS TO RECEIVE FILE
0628	CD3B09	CALL	ANS		;WAITS FOR ANSWER
062B	CD7809	CALL	XMIT		;TRANSMITS FILE
062E	CD3B09	CALL	ANS		
0631	CD280A	CALL	FILE		; "FILES" FILE IN CMS
0634	CD3B09	CALL	ANS		
0637	CD4608	CALL	TALLY		;PRINTS OUT RECORD COUNT
063A	C32004	JMP	TX		;RETURNS TO TRANSMIT MODE

;SUBR PROMPTS CONSOLE FOR FILE TO BE RECEIVED, SETS UP FILE
;CONTROL BLOCK AND CREATES FILE ON FLOPPY DISK, RECEIVES FILE
;FROM CMS AND ECHOES ON CONSOLE, CLOSES FILE AND RESTORES
;USER TO DIRECT CMS LINKUP
;

FILERX:

063D	3E00	MVI	A,	0	
063F	320301	STA	COUNT		
0642	320401	STA	COUNT+1		
0645	CD1C07	CALL	CPNAME		
0648	CD7D07	CALL	CRLF		
064B	CD6906	CALL	RESTR		;SETS UP FILE CONTROL BLOCK
064E	CD7D07	CALL	CRLF		
0651	CD5A07	CALL	MAKE		;DELETES AND CREATES DISK FILE
0654	CDEB08	CALL	BETA		;PREPARES CMS TO TRANSMIT FILE

069A CAA506	JZ	BONE		
069D C31307	JMP	REPEAT		
AONE:				
06A0 1E00	MVI	E,	0	
06A2 C3AA06	JMP	DSK		
BONE:				
06A5 1E01	MVI	E,	1	
06A7 C3AA06	JMP	DSK		
DSK:				
06AA 0E0E	MVI	C,	14	;CHANGES DISK DRIVE SELECTION
06AC CD0500	CALL	BDOS		
06AF 0E01	MVI	C,	1	
06B1 CD0500	CALL	BDOS		
06B4 FE3A	CPI	;		
06B6 C21307	JNZ	REPEAT		
06B9 0609	MVI	B,	9	;NEXT CHAR MUST BE " : "
06BB 21EB05	LXI	H,	FCB2+1	;IF NOT, START OVER
FNAME:				
06BE C5	PUSH	B		
06BF E5	PUSH	H		
06C0 0E01	MVI	C,	1	
06C2 CD0500	CALL	BDOS		
06C5 E1	POP	H		
06C6 C1	POP	B		
06C7 FE03	CPI	CNTLC		
06C9 CA0000	JZ	00		
06CC FE04	CPI	CNTLD		
06CE CA4F07	JZ	DIRECT		
06D1 FE15	CPI	CNTLU		
06D3 CA0D07	JZ	DUMMY		
06D6 FE2E	CPI	;		
06D8 CAE406	JZ	FTYPE		
06DB 77	MOV	M,	A	

06DC 23	INX H				
06DD 05	DCR B				; IF FILENAME EXCEEDS 8 CHAR, ; START OVER
06DE CA1307	JZ REPEAT				
06E1 C3BE06	JMP FNAME				
FTYPE:					
06E4 0604	MVI B,	4			
06E6 21F305	LXI H,	FCB2+9			
06E9 C5	PUSH B				
06EA E5	PUSH H				
06EB 0E01	MVI C,	1			
06ED CD0500	CALL BDOS				
06F0 E1	POP H				
06F1 C1	POP B				
06F2 FE33	CPI CNTLC				
06F4 CA0000	JZ 00				
06F7 FE04	CPI CNTLD				
06F9 CA4F07	JZ DIRECT				
06FC FE15	CPI CNTLU				
06FE CA0D07	JZ DUMMY				
0701 FE0D	CPI CR				
0703 C8	RZ				
0704 77	MOV M,	A			
0705 23	INX H				
0706 05	DCR B				; IF FILETYPE EXCEEDS 3 CHAR, ; START OVER
0707 CA1307	JZ REPEAT				
070A C3E906	JMP FTYPE1				
070D CD7D07	CALL CRLF				
0710 C36906	JMP RESTRT				
0713 11F602	LXI D,	MSG4			; PROMPTS "REPEAT"
0716 CDAC07	CALL MESSAGE				

075F CD0500	CALL	BDOS			
0762 0E16	MVI	C,	22		
0764 11EA05	LXI	D,	FCB2		
0767 CD0500	CALL	BDOS			
076A FEFF	CPI	255			
076C CA7407	JZ	NOROOM			
076F AF	XRA	A			
0770 320A06	STA	FCB2+32			
0773 C9	RET				
0774 114C03	LXI	D,	MSG11		
0777 CDAC07	CALL	MESSAGE			
077A C32004	JMP	TX			
077D 0E02	MVI	C,	2		
077F 1E0D	MVI	E,	CR		
0781 CD0500	CALL	BDOS			
0784 0E02	MVI	C,	2		
0786 1E0A	MVI	E,	LF		
0788 CD0500	CALL	BDOS			
078B C9	RET				
078C 11EA05	LXI	D,	FCB2		
078F 0E0F	MVI	C,	15		
0791 CD0500	CALL	BDOS			
0794 FEFF	CPI	255			
0796 CAA107	JZ	BADF			
0799 AF	XRA	A			
079A 320A06	STA	FCB2+32			
079D CD7D07	CALL	CRLF			
07A0 C9	RET				

```

;NEW FCB
;CREATES NEW FILE NAMED ABOVE

;ZERO INDICATES FULL DISK
;ZEROES FILE RECORD COUNTER

;PROMPTS "DISK FULL"

;STARTS NEW LINE ON CONSOLE

;OPENS DISK FILE FOR READING

;ZERO INDICATES NO SUCH FILE
;ZEROES FILE RECORD COUNTER

```

NOROOM:

CRLF:

OPEN:

07A1 110703	BADF:	LXI	D,	MSG5A	; PROMPTS "FILE NOT FOUND"
07A4 CDAC07		CALL	MESSAGE		
07A7 33		INX	SP		; ADJUSTS STACK POINTER
07A8 33		INX	SP		
07A9 C32004		JMP	TX		; RETURNS TO TRANSMIT MODE
		; PRINTS MESSAGE AT ADDR IN DE ON CONSOLE			
	MESSAGE:				
07AC 0E09		MVI	C,	9	
07AE CD0500		CALL	BDOS		
07B1 C9		RET			
		; READS ENTIRE DISK FILE INTO RAM STARTING AT			
		; BUFF (LIMITED TO 52K BYTES)			
	FILERD:				
	FILERD0:	LXI	H,	FLIMIT	
07B2 213002		SHLD	FCOUNT		
07B5 220501		LXI	D,	BUFF	
07B8 115A0A					
	FILERD1:	PUSH	D		
07BB D5		MVI	C,	26	
07BC 0E1A		CALL	BDOS		; CHANGES DMA BUFFER ADDR
07BE CD0500		LXI	D,	FCB2	
07C1 11EA05		MVI	C,	20	
07C4 0E14		CALL	BDOS		; READ FILE RECORD
07C6 CD0500		POP	D		
07C9 D1		PUSH	PSW		
07CA F5		CALL	COUNTER		
07CB CD7D08		LXI	H,	80H	; INCREMENTS BUFF BY 80H
07CE 218000		DAD	D		
07D1 19		XCHG			
07D2 EB		POP	PSW		
07D3 F1		CPI	0		
07D4 FE00		RNZ			; IF NOT ZERO, EOF CONTAINED IN
07D6 C0					; LAST RECORD
07D7 2A0501		LHLD	FCOUNT		

07DA 2B	DCX H				
07DB 220501	SHLD FCOUNT				
07DE 7C	MOV A, H				
07DF FE00	CPI 0				
07E1 C2BB07	JNZ FILERD1				
07E4 13	INX D				
07E5 3E13	MVI A, XOFF				
07E7 12	STAX D				
07E8 C9	RET				

;ZERO IF BUFFER EXCEEDED
 ;TEMPORARY EOF -- WILL TRANSMIT
 ;FIRST 52K BYTES OF FILE, THEN
 ;COME BACK TO READ MORE
 ;WRITES DISK FILE BY SAME ALGORITHM AS ABOVE
 FILEWR:

07E9 115A0A	LXI D, BUFF				
07EC 0680	MVI B, 80H				
07EE CD7D08	CALL COUNTER				
07F1 D5	PUSH D				
07F2 1A	LDAX D				
07F3 FE1A	CPI EOF				
07F5 CA1C08	JZ LAST				
07F8 13	INX D				
07F9 05	DCR B				
07FA C2F207	JNZ INLOOP2				
07FD D1	POP D				
07FE D5	PUSH D				
07FF 0E1A	MVI C, 26				
0801 CD0500	CALL BDOS				
0804 11EA05	LXI D, FCB2				
0807 0E15	MVI C, 21				
0809 CD0500	CALL BDOS				
080C D1	POP D				
080D F5	PUSH PSW				
080E 218000	LXI H, 80H				
0811 19	DAD D				

;MUST CHECK EACH RECORD FOR EOF
 ;IF EOF, THIS WILL BE LAST
 ;RECORD WRITTEN
 ;CHANGE DMA BUFFER ADDR
 ;WRITE ONE DISK RECORD
 ;INCREMENT BUFF BY 80H

0812 EB	XCHG					
0813 F1	POP	PSW				
0814 FE01	CPI	1				
0816 CA3008	JZ	ERR1				
0819 C3EC07	JMP	CONT				
	;WRITE LAST DISK RECORD					
	LAST:					
081C D1	POP	D				
081D 0E1A	MVI	C,	26			
081F CD0500	CALL	BDOS				
0822 11EA05	LXI	D,	FCB2			
0825 0E15	MVI	C,	21			
0827 CD0500	CALL	BDOS				
082A FE01	CPI	1				
082C CA3008	JZ	ERR1				
082F C9	RET					
	ERR1:					
0830 117803	LXI	D,	MSG13			
0833 CDAC07	CALL	MESSAGE				
0836 C9	RET					
	;CLOSES DISK FILE					
	CLOSE:					
0837 11EA05	LXI	D,	FCB2			
083A 0E10	MVI	C,	16			
083C CD0500	CALL	BDOS				
083F 112803	LXI	D,	MSG7			
0842 CDAC07	CALL	MESSAGE				
0845 C9	RET					
	;PRINTS OUT RECORD COUNT					
	TALLY:					
0846 3A0301	LDA	COUNT				
0849 1F	RAR					
084A 1F	RAR					
084B 1F	RAR					
084C 1F	RAR					
084D F60F	ANI	0FH				

;1 INDICATES DISK FULL

;PROMPTS "DISK FULL"

;PROMPTS "TRANSMISSION COMPLETE"

084F C630	ADI	30H	
0851 CD3705	CALL	CONOUT	
0854 3A0301	LDA	COUNT	
0857 E60F	ANI	0FH	
0859 C630	ADI	30H	
085B CD3705	CALL	CONOUT	
085E 3A0401	LDA	COUNT+1	
0861 1F	RAR		
0862 1F	RAR		
0863 1F	RAR		
0864 1F	RAR		
0865 E60F	ANI	0FH	
0867 C630	ADI	30H	
0869 CD3705	CALL	CONOUT	
086C 3A0401	LDA	COUNT+1	
086F E60F	ANI	0FH	
0871 C630	ADI	30H	
0873 CD3705	CALL	CONOUT	
0876 118403	LXI	D,	MSG14
0879 CDAC07	CALL	MESSAGE	
087C C9	RET		

;KEEPS TRACK OF RECORDS READ/WRITTEN
 COUNTER:

087D 3A0401	LDA	COUNT+1	
0880 C601	ADI	1	
0882 27	DAA		
0883 320401	STA	COUNT+1	
0886 3A0301	LDA	COUNT	
0889 CE00	ACI	0	
088B 320301	STA	COUNT	
088E C9	RET		

;RECEIVES WORDS FROM LINE USART AND STORES AT BUFF
 HAUL:

088F 116C03	LXI	D,	MSG12	;PROMPTS "RECEIVING"
0892 CDAC07	CALL	MESSAGE		
0895 11580A	LXI	D,	BUFF-2	;FIRST TWO WORDS WILL BE CR, LF


```

0898 0100D0
089B 3E13
089D CD6F05

08A0 DBE1
08A2 E602
08A4 CAA008
08A7 DB60
08A9 FE11
08AB CAD508
08AE FE13
08B0 CAA008
08B3 FE7F
08B5 CAA008
08B8 CD3705
08BB 12
08BC 13
08BD 0B
08BE 78
08BF FE00
08C1 CAE108
08C4 CDCA08
08C7 C3A008

08CA DBF7
08CC E602
08CE C8
08CF 310D04
08D2 C3A904

LXI B, BUFFMAX
MVI A, XOFF
CALL SEND
;CHECK USART FOR CHARACTER
FRX1:
IN 61H
ANI 2
JZ FRX1
IN 60H
CPI XON
JZ MARK
CPI XOFF
JZ FRX1
CPI 7FH
JZ FRX1
CALL CONOUT
STAX D
INX D
DCX B
MOV A, B
CPI 0
JZ EXCEED
CALL BREAK2
JMP FRX1
;CHECK KEYBOARD FOR INTERRUPT
;IF INTERRUPT EXISTS, RESET STACK POINTER
;AND JUMP TO DIRECT LINKUP MODE
;WHERE INTERRUPT CONDITION WILL BE NOTED
;AND A SIGNAL SENT TO CMS
BREAK2:
IN 0F7H
ANI 2
RZ
LXI SP, STKBTM
JMP CRCV1

;DON'T WANT THEM ON DISK
;BUFF LIMIT IS 52K BYTES
;TELL CMS TO START SENDING

;GET CHAR
;IF XON, THIS IS END OF FILE
;FILTER OUT XOFF AT END OF LINE
;FILTER OUT "NULLS"
;PRINT ON CONSOLE
;STORE IN BUFFER
;INCREMENT BUFFER ADDR
;CHECK BUFFER LIMIT NOT EXCEEDED

;LOOP FOREVER

```



```

;MARK END OF FILE WITH "EOF"
;LAST CHARS RECEIVED ARE CR,LF,NULL,R;>
;WANT TO BACK UP TO LAST VALID WORD
MARK:
08D5 1A LDAX D
08D6 1B DCX D
08D7 FE52 CPI 'R'
08D9 C2D508 JNZ MARK
08DC 13 INX D
08DD 3E1A MVI A, EOF
08DF 12 STAX D
08E0 C9 RET

EXCEED:
08E1 11B503 LXI D, MSG17 ;PROMPTS "BUFFER LIMIT EXCEEDED"
08E4 CDAC07 CALL MESSAGE
08E7 3E1A MVI A, EOF ;MARKS END OF FILE-REMAINDER OF
;FILE IS LOST

;SENDS "PRINT" TO CMS
BETA:
08E9 12 STAX D
08EA C9 RET

;SENDS "FILENAME FILETYPE" TO CMS
DELTA:
08EB 114503 LXI D, MSG10
08EE 1A LDAX D
08EF FE24 CPI '$'
08F1 CAFE08 JZ DELTA
08F4 CD3705 CALL CONOUT
08F7 CD6F05 CALL SEND
08FA 13 INX D
08FB C3EE08 JMP GAMMA

;SENDS "FILENAME FILETYPE" TO CMS
DELTA:
08FE 11440A LXI D, BUFF40
0901 1A LDAX D
0902 FE24 CPI '$'

```



```

0904 C8
0905 CD3705
0908 CD6F05
090B 13
090C C30109

RZ
CALL CONOUT
CALL SEND
INX D
JMP EPSILON
;SETS UP CMS TO RECEIVE FILE BY COMMANDING
;EDIT FILENAME FILETYPE
CMS:

090F 110103
LXI D, MSG5
CMS2:
0912 1A
LDAX D
0913 FE24
CPI '$'
0915 CA2209
JZ CMS3
0918 CD3705
CALL CONOUT
091B CD6F05
CALL SEND
091E 13
INX D
091F C31209
JMP CMS2
CMS3:
0922 11440A
LXI D, BUFF40
CMS4:
0925 1A
LDAX D
0926 FE24
CPI '$'
0928 CA3509
JZ CMS5
092B CD3705
CALL CONOUT
092E CD6F05
CALL SEND
0931 13
INX D
0932 C32509
JMP CMS4
CMS5:
0935 3E13
MVI A, XOFF
0937 CD6F05
CALL SEND
093A C9
RET
;ECHOES CMS ANSWER TO CONSOLE
ANS:
093B DB61
IN 61H
093D E602
ANI 2
093F CA3B09
JZ ANS

```



```

0942 DB60      IN      60H
0944 FE11      CPI      XON
0946 C8        RZ
0947 FE13      CPI      XOFF
0949 CA3B09    JZ      ANS
094C CD3705    CALL     CONOUT
094F C33B09    JMP      ANS

```

```

;FILTERS OUT XOFF

```

```

;RECEIVES CMS ANSWERS AND ECHOES TO CONSOLE
;FILTERS OUT XOFF,CR,LF,AND >
ANS2:

```

```

0952 DB61      IN      61H
0954 E602      ANI      2
0956 CA5209    JZ      ANS2
0959 DB60      IN      60H
095B FE11      CPI      XON
095D C8        RZ
095E FE13      CPI      XOFF
0960 CA5209    JZ      ANS2
0963 FE0D      CPI      CR
0965 CA5209    JZ      ANS2
0968 FE0A      CPI      LF
096A CA5209    JZ      ANS2
096D FE3E      CPI      '>'
096F CA5209    JZ      ANS2
0972 CD3705    CALL     CONOUT
0975 C35209    JMP      ANS2

```

```

;TRANSMITS FILE TO CMS
XMIT:

```

```

0978 111903    LXI      D,MSG6
097B CDAC07    CALL     MESSAGE
097E CD1D0A    CALL     PAUSE
0981 115A0A    LXI      D,BUFF
0984 0E83      MVI      C,83H
0986 1A        LDAX     D
0987 FE1A      CPI      EOF

```

```

;PROMPTS "TRANSMITTING"

```

```

; DELAY 100 MICROSECS AT
;BEGINNING OF EACH LINE
;132 BYTES

```

```

;IF EOF, TRANSMISSION FINISHED

```


0989 CAAF09	JZ	XMIT3			
098C FE13	CPI	XOFF			;IF TEMPORARY EOF, MORE DISK
					;FILE REMAINS
098E CAC909	JZ	XMIT4			
0991 FE0D	CPI	CR			;CLOSE OUT LINE AT CARR RETURN
0993 CA080A	JZ	ENDLN			
0996 FE0A	CPI	LF			;FILTER OUT LINEFEEDS
0998 CAA809	JZ	SKIP			
099B FE09	CPI	09H			;CHANGE TAB CHAR TO "?"
099D CC7904	CZ	CHNG1			
09A0 47	MOV	B,	A		
09A1 CD6F05	CALL	SEND			
09A4 0D	DCR	C			
09A5 CA0C0A	JZ	ENDLN2			;IF 132 CHARS EXCEEDED, CMS
					;BUFFER CHOKES
	SKIP:				
09A8 13	INX	D			
09A9 CDF909	CALL	BREAK3			
09AC C38609	JMP	XMIT2			
	XMIT3:				
09AF 112803	LXI	MSG7			;PROMPTS "TRANSMISSION COMPLETE"
09B2 CDAC07	CALL	MESSAGE			
	XMIT35:				
09B5 CD1D0A	CALL	PAUSE			
09B8 3E13	MVI	A,	XOFF		;SENDS DOUBLE XOFF TO SHIFT
09BA CD6F05	CALL	SEND			;CMS FROM INPUT TO EDIT MODE
09BD CD5209	CALL	ANS2			;WAIT FOR ANSWER AND DELAY
09C0 CD1D0A	CALL	PAUSE			
09C3 3E13	MVI	A,	XOFF		
09C5 CD6F05	CALL	SEND			
09C8 C9	RET				

;FOR FILES EXCEEDING 52K, PROGRAM SHIFTS
 ;CMS TO EDIT MODE AND ISSUES "SAVE" COMMAND
 ;AT THIS POINT - CMS SAVES TRANSMITTED DATA
 ;AND RETURNS TO INPUT MODE, AT WHICH TIME
 ;PROGRAM READS NEXT SECTION OF FILE AND TRANSMITS

09C9	CD8509	CALL	XMIT35	
09CC	CD3B09	CALL	ANS	
09CF	CD1D0A	CALL	PAUSE	
09D2	11F403	LXI	D,	MSG19
09D5	1A	LDAX	D	
09D6	FE24	CPI	'\$'	
09D8	CAE509	JZ	XMIT6	
09DB	CD3705	CALL	CONOUT	
09DE	CD6F05	CALL	SEND	
09E1	13	INX	D	
09E2	C3D509	JMP	XMIT5	
09E5	3E13	MVI	A,	XOFF
09E7	CD6F05	CALL	SEND	
09EA	CD3B09	CALL	ANS	
09ED	11E803	LXI	D,	MSG18
09F0	CDAC07	CALL	MESSAGE	
09F3	CD8207	CALL	FILERD0	
09F6	C37809	JMP	XMIT	
09F9	DBF7	IN	0F7H	
09FB	E602	ANI	2	
09FD	C8	RZ		
09FE	DBF6	IN	0F6H	
0A00	E67F	ANI	7FH	
0A02	FE04	CPI	CNTLD	
0A04	C0	RNZ		
0A05	C34F07	JMP	DIRECT	
		;SENDS XOFF AFTER EACH LINE		
ENDLN:				
0A08	B8	CMP	B	
0A09	CAA809	JZ	SKIP	
0A0C	47	MOV	B,	A

;IF LAST CHAR WAS A CR, IGNORE-
;CANCELS SKIPPED LINES

0A0D 3E13	MVI A, XOFF	
0A0F CD6F05	CALL SEND	
0A12 CD5209	CALL ANS2	
0A15 CD1D0A	CALL PAUSE	
0A18 0E83	MVI C, 83H	;132 BYTES
0A1A C3A809	JMP SKIP	;CONTINUE TRANSMITTING
	;DELAY APPROX 100 MICROSECONDS	
	PAUSE:	
0A1D 210002	LXI H, 200H	
	PAUSE2:	
0A20 2B	DCX H	
0A21 7C	MOV A, H	
0A22 FE00	CPI 0	
0A24 C2200A	JNZ PAUSE2	
0A27 C9	RET	
	;COMMANDS CMS TO "FILE" TRANSMITTED DATA	
	FILE:	
0A28 CD1D0A	CALL PAUSE	
0A2B 114003	LXI D, MSG8	
	FILE2:	
0A2E 1A	LDAX D	
0A2F FE24	CPI '\$'	
0A31 CA3E0A	JZ FILE3	
0A34 CD3705	CALL CONOUT	
0A37 CD6F05	CALL SEND	
0A3A 13	INX D	
0A3B C32E0A	JMP FILE2	
	FILE3:	
0A3E 3E13	MVI A, XOFF	
0A40 CD6F05	CALL SEND	
0A43 C9	RET	
0A44	BUFF40: DS 20	
0A58	DS 2	
0A5A =	BUFF EQU \$	
0A5A	END 100H	;BUFFER STARTS AT END OF PROGRAM

APPENDIX H

```

;UPDATED 14 AUG 78      VERS 73

ORG 100H
JMP START
BDOS EQU
XON EQU
XOFF EQU
CR EQU
LF EQU
FF EQU
EOF EQU
BUFF EQU
CONV EQU
RUB EQU
CNTLC EQU
CNTLD EQU
CNTLG EQU
CNTLI EQU
CNTLP EQU
CNTLR EQU
CNTLT EQU
CNTLU EQU
PPREG: DS

5H      ;ENTRY POINT
11H      ; END OF LINE FROM VIRTUAL MACHINE
13H      ;END OF LINE TO VIRTUAL MACHINE
0DH      ;CARRIAGE RETURN
0AH      ;LINE FEED
0CH      ;FORM FEED
1AH      ;END OF FILE CHAR FOR DISK WRITE
880H     ;START OF MEMORY BUFFER
0FE0EH   ;MONITOR CONVERSION ROUTINE
7FH      ;DELETE CHARACTER
03H      ;WARM BOOT
04H      ;RESTORES "DIRECT LINKUP" MODE
07H      ;PRINT INSTRUCTIONS
09H      ;TAB CHARACTER
10H      ;CONTROL P TURNS PRINTER ON AND OFF
12H      ;CONTROL R FOR RECEIVE FILE
14H      ;CONTROL T FOR TRANSMIT FILE
15H      ;DELETE LINE
1        ;PRINTER CONTROL REG;0 OFF,1 ON

```



```

MSG1: DB
MSG2: DB
MSG3: DB
MSG4: DB
MSG5: DB
MSG5A: DB
MSG6: DB
MSG7: DB
MSG8: DB
MSG9: DB
MSG10: DB
MSG11: DB
MSG12: DB
MSG15: DB
STACK: DS
STKBTM EQU

CR,LF,'DIAL 2721 FOR LINE
CONTROL C -- CONTROL G FOR INSTRUCTIONS',CR,LF,'$'
CONTROL D - REBOOT',CR,LF
CONTROL G - RETURN TO DIRECT LINKUP',CR,LF
CONTROL I - INSTRUCTIONS',CR,LF
CONTROL I - TAB',CR,LF
CONTROL P - PRINTER ON/OFF',CR,LF
CONTROL T - TRANSMIT FILE',CR,LF
CONTROL U - DELETE LINE',CR,LF
RUBOUT - DELETE CHARACTER',CR,LF
XMIT - INTERRUPT CMS',CR,LF,'$'
CR,LF,'DISK:FILENAME.FILETYPE',CR,LF,'$'
CR,LF,'REPEAT',CR,LF,'$'
EDIT $
FILE NOT FOUND',CR,LF,'>$'
TRANSMITTING',CR,LF,'$'
TRANSMISSION COMPLETE',CR,LF,'$'
FILE$
DATA POINTS',CR,LF,'$'
TO RECEIVE FILE, USE LINK PROGRAM',CR,LF,'$'
SCANS PER SECOND',CR,LF,'$'
RUN CONTROL NUMBER $
CMS FILENAME FILETYPE?',CR,LF,'$'
20
$

```

```

START:
LXI SP, STKBTM
MVI A, 0
STA PPREG
LXI D, MSG1
CALL MESSAGE
CALL BOARD
IN 60H
;
;INITIALLY PRINTER IS OFF
;PROMPTS USER TO CALL FOR LINE
;INITIALIZES SBC 534 BOARD
;TRANSMIT MODE

```


; TX:

03B3 DB61	IN	61H	;CHECKS LINE FOR MESSAGE
03B5 E602	ANI	2	
03B7 C23C04	JNZ	CRCV1	
03BA DBF7	IN	0F7H	;CHECKS KEYBOARD
03BC E602	ANI	2	
03BE CAB303	JZ	TX	;LOOPS UNTIL ONE OF THE ABOVE
03C1 0E01	MVI	C,	
03C3 CD0500	CALL	BDOS	;READ CHAR FROM CONSOLE
03C6 FE0D	CPI	CR	;CHECK FOR CR
03C8 CA2504	JZ	RCV	;SWITCH TO RECEIVE MODE
03CB FE10	CPI	CNTLP	
03CD CA9604	JZ	PRTCONT	;TURN PRINTER ON/OFF
03D0 FE12	CPI	CNTLR	
03D2 CAC805	JZ	FILERX	;RECEIVE FILE MODE
03D5 FE14	CPI	CNTLT	
03D7 CA9E05	JZ	FILETX	;TRANSMIT FILE MODE
03DA FE03	CPI	CNTLC	
03DC CA0000	JZ	00H	;ESCAPE BY REBOOTING
03DF FE07	CPI	CNTLG	;PRINT INSTRUCTIONS
03E1 CA7904	JZ	GOUGE	
03E4 FE09	CPI	CNTLI	
03E6 CC1F04	CZ	CHNG4	;TRANSMIT TAB CHAR "?"
03E9 FE7F	CPI	RUB	
03EB CC0F04	CZ	CHNG2	;TRANSMIT DELETE CHAR SYMBOL "Q"
03EE FE15	CPI	CNTLU	
03F0 CA1704	JZ	CHNG3	;TRANSMIT DELETE LINE SYMBOL "["
			;AND XOFF
03F3 4F	MOV	C,	
03F4 FE11	CPI	XON	
03F6 CA0504	JZ	CTX	
03F9 3A0301	LDA	PPREG	;CHECK IF PRINTER ON
03FC FE00	CPI	0	
03FE CA0504	JZ	CTX	
0401 79	MOV	A,	

0402	CDBF04	CALL	DRIVER		
0405	79	MOV	A,	C	
0406	CD0205	CALL	SEND		
0409	C3B303	JMP	TX		;SENDS CHAR TO VIRTUAL MACHINE ;LOOPS FOREVER
040C	3E3F	MVI	A,	'?'	
040E	C9	RET			
040F	3E08	MVI	A,	08H	
0411	CDCA04	CALL	CONOUT		;BACKSPACE
0414	3E40	MVI	A,	'@'	
0416	C9	RET			
0417	3E5B	MVI	A,	'['	
0419	CD0205	CALL	SEND		
041C	C32504	JMP	RCV		
041F	3E3F	MVI	A,	'?'	
0421	CDCA04	CALL	CONOUT		
0424	C9	RET			
;RECEIVE MODE					
;RCV:					
0425	3A0301	LDA	PPREG		;CHECK IF PRINTER ON
0428	FE00	CPI	0		
042A	CA3704	JZ	CRCV		
042D	3E0D	MVI	A,	CR	
042F	CDBF04	CALL	DRIVER		;START NEW LINE ON PRINTER
0432	3E0A	MVI	A,	LF	
0434	CDBF04	CALL	DRIVER		
0437	3E13	MVI	A,	XOFF	
0439	CD0205	CALL	SEND		;END OF LINE CHAR
CRCV1:					

0475 13	BACK:	INX	D		
0476 C36004		JMP	LOOP		;LOOP UNTIL CAUGHT UP
0479 113901	GOUGE:	LXI	D,	MSG2	
047C 1A	GLOOP:	LDAX	D		
047D FE24		CPI	'\$'		
047F CAB303		JZ	TX		
0482 CDCA04		CALL	CONOUT		
0485 47		MOV	B,	A	
0486 3A0301		LDA	PPREG		
0489 FE00		CPI	0		
048B CA9204		JZ	GLP		
048E 78		MOV	A,	B	
048F CDBF04	GLP:	CALL	DRIVER		
0492 13		INX	D		
0493 C37C04		JMP	GLOOP		
0496 3A0301	PRTCONT:	LDA	PPREG		;CHECK IF PRINTER ON OR OFF
0499 FE00		CPI	0		
049B C2B304		JNZ	PRTOFF		;IF ON, WANT TO TURN OFF
049E CD7805		CALL	USART2		
04A1 3E01		MVI	A,	1	
04A3 320301		STA	PPREG		;LATER ROUTINES CHECK THIS ADDR
04A6 3E0D		MVI	A,	CR	
04A8 CDBF04		CALL	DRIVER		
04AB 3E0A		MVI	A,	LF	
04AD CDBF04		CALL	DRIVER		
04B0 C3B303	PRTOFF:	JMP	TX		;RETURN TO TRANSMIT MODE
04B3 3E30		MVI	A,	30H	
04B5 D363		OUT	63H		;CONTROL WORD TO TURN PRINT OFF
04B7 3E00		MVI	A,	0	
04B9 320301		STA	PPREG		;LATER ROUTINES CHECK THIS ADDR

04BC C3B303	JMP TX		
	;ROUTINE TO DRIVE PRINTER USART		
	DRIVER:	PUSH PSW	
04BF F5			
	SLO:		
04C0 DB63	IN	63H	
04C2 0F	RRC		
04C3 D2C004	JNC	SLO	
04C6 F1	POP	PSW	
04C7 D362	OUT	62H	
04C9 C9	RET		
			;WAIT UNTIL XMITTER READY
	;ROUTINE TO DRIVE CONSOLE USART		
	CONOUT:	PUSH PSW	
04CA F5			
	SLO2:		
04CB DBF7	IN	0F7H	
04CD 0F	RRC		
04CE D2CB04	JNC	SLO2	
04D1 F1	POP	PSW	
04D2 D3F6	OUT	0F6H	
04D4 C9	RET		
			;KEEPS TRACK OF WHICH DATA HAS BEEN PRINTED
	CKPRT:		
04D5 7D	MOV	A, L	
04D6 BB	CMP	E	
04D7 CA4204	JZ	RX1	
04DA DBF7	IN	0F7H	
04DC 0F	RRC		
04DD D24204	JNC	RX1	
			;CAUGHT UP - NO NEED TO PROCEED
04E0 3A0301	LDA	PPREG	
04E3 FE00	CPI	0	
04E5 CAEE04	JZ	CKP2	
			;CONSOLE NOT READY - NO NEED TO PROCEED
			;CHECK PRINTER ON
04E8 DB63	IN	63H	
			;PRINTER NOT ON - NO NEED TO PROCEED


```

; ; TWO MORE USARTS AND ONE 8255 PARALLEL INTERFACE AND THEIR TIMERS ARE
; ; AVAILABLE ON THE 534 BOARD. NEW INTERFACES MUST BE PROGRAMMED BEFORE USE
; ;
; ;
; ; DI OUT 6FH
; ; 0541 F3 ; DISABLES 8080 INTERRUPTS
; ; 0542 D36F ; RESETS BOARD
; ; 0544 D36C ; SELECTS BOARD CONTROL BLOCK
; ; 0546 CD4E05 CALL TIMER
; ; 0549 CD6B05 CALL USART
; ; 054C FB EI
; ; 054D C9 RET

```



```

;
;
;
;SET UP BOTH USARTS WITH RESETS AND MODE WORDS
;
;
USART:
056B 3ECA          MVI    A,      0CAH      ;2 STOP, PAR DISABLED, 7 BITS
056D D361          OUT    A,      61H
056F 3E5A          MVI    A,      5AH      ;1 STOP, PAR DISABLED, 7 BITS
0571 D363          OUT    A,      63H
0573 3E37          MVI    A,      37H
0575 D361          OUT    A,      61H
0577 C9            RET

USART2:
0578 3E33          MVI    A,      33H
057A D363          OUT    A,      63H
057C C9            RET

;THIS SECTION PERTAINS TO TRANSFERRING COMPLETE
;FILES BETWEEN MDS AND IBM 360
;
FCB EQU 5CH
FCBCN EQU FCB+0
FCBFN EQU FCB+1
FCBFT EQU FCB+9
FCBRL EQU FCB+12
FCBRC EQU FCB+15
FCB2: DS 33
FCBCR EQU FCB+32
;FCB ADDR
;DISK NAME
;FILENAME(8CHAR)
;FILETYPE (3CHAR)
;REEL NUMBER
;FILE RECORD COUNT (0-127)
;NEW FILENAME AND FILETYPE
;NEXT RECORD NUMBER

;SUBR PROMPTS CONSOLE FOR FILE TO BE XMITTED, SETS UP FILE
;CONTROL BLOCK, OPENS NEW CMS FILE, TRANSMITS FILE, AND
;RETURNS USER TO DIRECT CMS LINKUP
;
FILETX:
059E CDD105        CALL    RESTRT      ;SETS UP FILE CONTROL BLOCK

```


05A1 CDC706	CALL CRLF	CP/CMS FILENAME, FILETYPE
05A4 CD8906	CALL CPNAME	
05A7 CDC706	CALL CRLF	
05AA CDD206	CALL OPEN	;OPENS DISK FILE
05AD CD0307	CALL FILERD	;READS DISK FILE
05B0 CD2207	CALL ECHO	;ECHO FILE INFO
05B3 CD5F07	CALL CMS	;PREPARES CMS TO RECEIVE FILE
05B6 CD8B07	CALL ANS	;WAITS FOR ANSWER
05B9 CDC807	CALL XMIT	;TRANSMITS FILE
05BC CD8B07	CALL ANS	
05BF CD3C08	CALL FILE	;"FILES" FILE IN CMS
05C2 CD8B07	CALL ANS	
05C5 C3B303	JMP TX	;RETURNS TO TRANSMIT MODE

THIS PROGRAM DOES NOT HAVE RECEIVE FILE MODE
 FILERX:

05C8 112703	LXI D, MSG10
05CB CDF206	CALL MESSAGE
05CE C3B303	JMP TX

CLEAR OUT OLD FILE CONTROL BLOCK AND SETS UP NEW ONE
 RESTRT:

05D1 11AE02	LXI D, MSG3	PROMPTS "FILENAME.FILETYPE"
05D4 CDF206	CALL MESSAGE	
05D7 3E00	MVI A, 0	
	STA FCB2	;PADS NEW FCB
05DC 217E05	LXI H, FCB2+1	
05DF 3E20	MVI A, 20H	;BLANK CHAR
05E1 060B	MVI B, 11	

05E3 77	MOV M, A
05E4 23	INX H
05E5 05	DCR B
05E6 C2E305	JNZ PAD1

05E9 3E00	MVI A,	0	
05EB 0604	MVI B,	4	
05ED 218905	LXI H,		FCB2+12
PAD2:			
05F0 77	MOV M,	A	
05F1 23	INX H		
05F2 05	DCR B		
05F3 C2F005	JNZ PAD2		
05F6 0E01	MVI C,	1	
05F8 CD0500	CALL BDOS		
05FB FE41	CPI 'A'		:ASKS FOR DESIRED DISK ;AND NOTIFIES DISK DRIVE
05FD CA0D06	JZ AONE		
0600 FE42	CPI 'B'		
0602 CA1206	JZ BONE		
0605 FE04	CPI CNTLD		
0607 CAB006	JZ DIRECT		
060A C3E006	JMP REPEAT		
060D 1E00	MVI E,	0	
060F C31706	JMP DSK		
0612 1E01	MVI E,	1	
0614 C31706	JMP DSK		
0617 0E0E	MVI C,	14	
0619 CD0500	CALL BDOS		
061C 0E01	MVI C,	1	
061E CD0500	CALL BDOS		
0621 FE3A	CPI ':'		:NEXT CHAR MUST BE ":"
0623 C28006	JNZ REPEAT		:IF NOT, START OVER
0626 0609	MVI B,	9	
AONE:			
BONE:			
DSK:			:CHANGES DISK DRIVE SELECTION

0628	217E05	LXI	H,	FCB2+1
FNAME:				
062B	C5	PUSH	B	
062C	E5	PUSH	H	
062D	0E01	MVI	C,	1
062F	CD0500	CALL	BDOS	
0632	E1	POP	H	
0633	C1	POP	B	
0634	FE03	CPI	CNTLC	
0636	CA0000	JZ	00	
0639	FE04	CPI	CNTLD	
063B	CABC06	JZ	DIRECT	
063E	FE15	CPI	CNTLU	
0640	CA7A06	JZ	DUMMY	
0643	FE2E	CPI	'.	
0645	CA5106	JZ	FTYPE	
0648	77	MOV	M,	A
0649	23	INX	H	
064A	05	DCR	B	
064B	CA8006	JZ	REPEAT	
064E	C32B06	JMP	FNAME	

;IF FILENAME EXCEEDS 8 CHAR,
;START OVER

0651	0604	MVI	B,	4
0653	218605	LXI	H,	FCB2+9
FTYPE:				
FTYPE1:				
0656	C5	PUSH	B	
0657	E5	PUSH	H	
0658	0E01	MVI	C,	1
065A	CD0500	CALL	BDOS	
065D	E1	POP	H	
065E	C1	POP	B	
065F	FE03	CPI	CNTLC	
0661	CA0000	JZ	00	
0664	FE04	CPI	CNTLD	
0666	CABC06	JZ	DIRECT	

ADDRESS	OPERATION	OPERAND	COMMENT
0669	FE15		
066B	CA7A06		
066E	FE0D		
0670	C8		
0671	77		
0672	23		
0673	05		
0674	CA8006		
0677	C35606		
067A	CD0706		
067D	C3D105		
0680	11C902		
0683	CD0206		
0686	C3D105		
0689	117303		
068C	CD0206		
068F	115808		
0692	D5		
0693	0E01		
0695	CD0500		
0698	D1		
0699	FE03		
069B	CA0000		
069E	FE04		
06A0	CABC06		
06A3	FE15		
06A5	CAB606		
06A8	FE0D		
06AA	CAB206		
06AD	12		
06AE	13		
06AF	C39206		
CPI	CNTLU		
JZ	DUMMY		
CPI	CR		
RZ			
MOV	M,		
INX	H		
DCR	B		
JZ	REPEAT		
JMP	FTYPE1		
DUMMY:			
CALL	CRLF		
JMP	RESTR		
REPEAT:			
LXI	MSG4		
CALL	MESSAGE		
JMP	RESTR		
CPNAME:			
LXI	MSG15		
CALL	MESSAGE		
LXI	D,		
PUSH	D		
MVI	C,		
CALL	BDOS		
POP	D		
CPI	CNTLC		
JZ	00		
CPI	CNTLD		
JZ	DIRECT		
CPI	CNTLU		
JZ	DUMMY2		
CPI	CR		
JZ	NAME3		
STAX	D		
INX	D		
JMP	NAME2		

06B2 3E24	NAME3:	MVI	A,		
06B4 12		STAX	D		
06B5 C9		RET			
06B6 CDC706	DUMMY2:	CALL	CRLF		
06B9 C38906		JMP	CPNAME		
06BC 31A003	DIRECT:				
06BF 3E13		LXI	SP,	STKBTM	
06C1 CD0205		MVI	A,	XOFF	
06C4 C33C04		CALL	SEND		
06C7 3E0D		JMP	CRCV1		
06C9 CDCA04	CRLF:				
06CC 3E0A		MVI	A,	CR	
06CE CDCA04		CALL	CONOUT		
06D1 C9		MVI	A,	LF	
		CALL	CONOUT		
		RET			
06D2 117D05	OPEN:	LXI	D,	FCB2	
06D5 0E0F		MVI	C,	15	
06D7 CD0500		CALL	BDOS		
06DA FEFF		CPI	255		
06DC CAE706		JZ	BADF		
06DF AF		XRA	A		
06E0 329D05		STA	FCB2+32		
06E3 CDC706		CALL	CRLF		
06E6 C9		RET			
06E7 11DA02	BADF:				
06EA CDF206		LXI	D,	MSG5A	
06ED 33		CALL	MESSAGE		
06EE 33		INX	SP		
06EF C3B303		INX	SP		
		JMP	TX		

;OPENS DISK FILE FOR READING
 ;ZERO INDICATES NO SUCH FILE
 ;ZEROES FILE RECORD COUNTER
 ;PROMPTS "FILE NOT FOUND"
 ;ADJUSTS STACK POINTER
 ;RETURNS TO TRANSMIT MODE


```

;PRINTS MESSAGE AT ADDR IN DE ON CONSOLE

```

MESSAGE:

06F2 0E09
06F4 CD0500
06F7 C9

C.
B.DOS

9

MESS2:

06F8	1A
06F9	13
06FA	FE24
06FC	C8
06FD	CDCA04
0700	C3F806

```
LDAX D
INX D
CPI '5'
RZ
CALL CONOUT
JMP MESS2
```

```
;READS ENTIRE DISK FILE INTO RAM STARTING AT
;BUFF (LIMITED TO 52K BYTES)
```

FILED:

FILED:

0703 118008

LXI D. BUFF

FILERO1:

0706 D5
0707 0E1A
0709 CD0500
070C 117D05
070F 0E14
0711 CD0500
0714 D1
0715 F5
0716 218000
0719 19
071A EB
071B F1
071C FE00
071E C0
071F C30607

```

D
PUSH
C,
MVI
CALL
26
BDOS
;CHANGES DMA BUFFER ADDR
D,
LXI
FCB2
MVI
20
CALL
BDOS
;READ FILE RECORD

```

```

;ROUTINE TO ECHO FILE RECORD DATA TO CONSOLE
;-FIRST FILE RECORD (BUFF) CONTAINS "DATA1 (ASCII).
;START CHAN (HEX), FINAL CHAN (HEX), NUMBER DATA POINTS$
;SCAN RATE$RUN CONTROL NUMBER$ (ALL IN ASCII)

```



```

;BUFF+20H CONTAINS UPPER MEMORY LIMIT OF DATA ---
;BUFF+30H CONTAINS SCAN WORD LENGTH (EFFECTIVE
;LINE LENGTH FOR TRANSMISSION TO CMS)
;
ECHO:
0722 CDC706      CALL      CRLF
0725 118008      LXI      D,      BUFF
0728 0605        MVI      B,      6H
;SKIP LINE
;FIRST LINE OF FILE

ELOOP:
072A 1A          LDAX     D
072B CDCA04      CALL     CONOUT
072E 13          INX      D
072F 05          DCR      B
0730 C22A07      JNZ      ELOOP
0733 CDC706      CALL     CRLF
0736 118708      LXI      D,      BUFF+8H
0739 CDF806      CALL     MESS2
073C D5          PUSH     D
073D 111803      LXI      D,      MSG9
0740 CDF206      CALL     MESSAGE
0743 D1          POP      D
0744 CDF806      CALL     MESS2
0747 D5          PUSH     D
0748 114B03      LXI      D,      MSG11
074B CDF206      CALL     MESSAGE
074E 115F03      LXI      D,      MSG12
0751 CDF206      CALL     MESSAGE
0754 D1          POP      D
0755 CDF806      CALL     MESS2
0758 CDC706      CALL     CRLF
075B CDC706      CALL     CRLF
075E C9          RET
;SKIP LINE
;NUMBER OF DATA
;POINTS LOCATED HERE
;DATA POINTS"
;DE REG CONTAINS ADDR OF
;FILE PARAMETERS
;SCANS PER SECOND"
;RUN CONTROL NUMBER"

;SETS UP CMS TO RECEIVE FILE BY COMMANDING
;"EDIT FILENAME FILETYPE"
CMS:

```


075F 11D402		LXI	D,	MSG5
0762 1A	CMS2:	LDAX	D	
0763 FE24		CPI	'\$'	
0765 CA7207		JZ	CMS3	
0768 CDCA04		CALL	CONOUT	
076B CD0205		CALL	SEND	
076E 13		INX	D	
076F C36207		JMP	CMS2	
0772 115808	CMS3:	LXI	D,	BUFF40
0775 1A	CMS4:	LDAX	D	
0776 FE24		CPI	'\$'	
0778 CA8507		JZ	CMS5	
077B CDCA04		CALL	CONOUT	
077E CD0205		CALL	SEND	
0781 13		INX	D	
0782 C37507		JMP	CMS4	
0785 3E13	CMS5:	MVI	A,	XOFF
0787 CD0205		CALL	SEND	
078A C9		RET		
		;ECHOES CMS ANSWER TO CONSOLE		
	ANS:			
078B DB61		IN	61H	
078D E602		ANI	2	
078F CA8B07		JZ	ANS	
0792 DB60		IN	60H	
0794 FE11		CPI	XON	
0796 C8		RZ		
0797 FE13		CPI	XOFF	
0799 CA8B07		JZ	ANS	
079C CDCA04		CALL	CONOUT	
079F C38B07		JMP	ANS	
		;RECEIVES CMS ANSWERS AND ECHOES TO CONSOLE		
				;FILTERS OUT XOFF

;FILTERS OUT XOFF,CR,LF,AND >
ANS2:

07A2 DB61
07A4 E602
07A6 CAA207
07A9 DB60
07AB FE11
07AD C8
07AE FE13
07B0 CAA207
07B3 FE0D
07B5 CAA207
07B8 FE0A
07BA CAA207
07BD FE3E
07BF CAA207
07C2 CDCA04
07C5 C3A207

IN 61H
ANI 2
JZ ANS2
IN 60H
CPI XON
RZ
CPI XOFF
JZ ANS2
CPI CR
JZ ANS2
CPI LF
JZ ANS2
CPI '>'
JZ ANS2
CALL CONOUT
JMP ANS2

;TRANSMITS FILE TO CMS

XMIT:

07C8 11EC02
07CB CDF206
07CE CD3108

LXI
CALL
CALL

MSG6 D, MESSAGE
PAUSE

; PROMPTS "TRANSMITTING"

; DELAY 100 MICROSECS AT BEGIN-
; NING OF EACH LINE

07D1 110009
07D4 3AB008
07D7 67

LXI D, BUFF+80H
LDA BUFF+30H
MOV H, A

; NUMBER CHAR PER LINESCAN

XMIT2:

07D8 1A
07D9 CDFB07
07DC 78
07DD CD0205
07E0 79
07E1 CD0205
07E4 25
07E5 CA1808

LDAX D
CALL ASCII
MOV A, B
CALL SEND
MOV A, C
CALL SEND
DCR H
JZ ENDLN2

07E8 13	SKIP:	INX	D	
07E9 CD0908		CALL	BREAK3	
07EC C3D807		JMP	XMIT2	
07EF 11FB02	XMIT3:	LXI	MSG7	; PROMPTS "TRANSMISSION COMPLETE"
07F2 CDF206		CALL	MESSAGE	
07F5 3E13	XMIT35:	MVI	A, XOFF	; SENDS DOUBLE XOFF TO SHIFT
07F7 CD0205		CALL	SEND	; CMS FROM INPUT TO EDIT MODE
07FA C9		RET		; WAIT FOR ANSWER AND DELAY
07FB F5		; ROUTINE CONVERTS HEX BYTE TO TWO ASCII CHARS		
07FC 0F	ASCII:	PUSH	PSW	
07FD 0F		RRC		
07FE 0F		RRC		
07FF 0F		RRC		
0800 CD0EFE		CALL		
0803 41		MOV	C, CONV	; SAVES ONE IN B REG
0804 F1		POP	PSW	
0805 CD0EFE		CALL	CONV	; OTHER RETURNED IN C REG
0808 C9		RET		
0809 DBF7	; BREAK3:	IN	0F7H	
080B E602		ANI	2	
080D C8		RZ		
080E DBF6		IN	0F6H	
0810 E67F		ANI	7FH	
0812 FE04		CPI	CNTLD	
0814 C0		RNZ		
0815 C3BC06		JMP	DIRECT	
0818 3E13	ENDLN2:	MVI	A, XOFF	
081A CD0205		CALL	SEND	

081D CDA207	CALL	ANS2	
0820 CD3108	CALL	PAUSE	
0823 3AA008	LDA	BUFF+20H	;SEE IF DATA EXHAUSTED
0826 BA	CMP	D	
0827 CAEF37	JZ	XMIT3	
082A 3AB008	LDA	BUFF+30H	
082D 67	MOV	H, A	
082E C3E807	JMP	SKIP	;CONTINUE TRANSMITTING
;DELAY APPROX 100 MICROSECONDS			
0831 210002	LXI	H, 200H	
PAUSE2:			
0834 2B	DCX	H	
0835 7C	MOV	A, H	
0836 FE00	CPI	0	
0838 C23408	JNZ	PAUSE2	
083B C9	RET		
;COMMANDS CMS TO "FILE" TRANSMITTED DATA			
FILE:			
083C CD3108	CALL	PAUSE	
083F 111303	LXI	D, MSG8	
FILE2:			
0842 1A	LDAX	D	
0843 FE24	CPI	'\$'	
0845 CA5208	JZ	FILE3	
0848 CDCA04	CALL	CONOUT	
084B CD0205	CALL	SEND	
084E 13	INX	D	
084F C34208	JMP	FILE2	
FILE3:			
0852 3E13	MVI	A, XOFF	
0854 CD0205	CALL	SEND	
0857 C9	RET		
DS 20			
0857 C9			DS 2
BUFF40:			


```

INDEX: DS
MODE: DS
NEAT: DS
SKNDEX: DS
TEXT: DS
TYTLE: DS
STACK: DS
STKBTM EQU

1 ;LINE SPACER INDEX
1 ;ALL OR PARTIAL MODE INDEX
2 ;INDEX FOR BLANKING FIRST LINE
1 ;LINE SKIP INDEX
1 ;1 IF TEXT FILE
12 ;TITLE WILL BE STORED HERE
64 ;RESERVE STACK SPACE
$

;MESSAGES
$,
MSG15: DB
MSG2: DB
MSG3: DB
MSG4: DB
MSG7: DB
MSG8: DB
MSG9: DB
MSG10: DB
MSG13: DB
MSG14: DB

'TEXT FILE?? (Y/N) $'
'TYPE 2 FOR DOUBLE SPACE $'
'(DEFAULT = SINGLE SPACE) $'
'FILE NOT FOUND $'
'CHECK FOR ERRORS IN CURRENT RECORD $'
'HAVE A NICE DAY $'
'DONE $'
'TYPE K TO CANCEL OR SPACE TO CONTINUE $'
'PRINT ALL (A) OR PART (P) ?? $'
'ENTER STRING1,STRING2 -- (LIMIT 15 CHARACTERS EACH) $'

;MAIN PROGRAM

MAIN: LXI SP, STKBTM
CALL BOARD
MVI A, 0H
STA MODE
STA TEXT
STA LCOUNT
;OPEN DISK FILE FOR READING
SETUP:
LXI D, FCB
MVI C, OPENF
CALL BDOS
;CHECK FOR ERRORS

```


028B FEFF	CPI	255
028D CA6C03	JZ	BADF
	;GOOD OPEN	
0290 AF	XRA	A
0291 327C00	STA	FCBCR
0294 CDCF04	CALL	CRLF
0297 116401	LXI	D, MSG15
029A CDED04	CALL	CRTMSG
029D 0E01	MVI	C, READC
029F CD0500	CALL	BDOS
02A2 321001	STA	TEXT
02A5 FE59	CPI	'Y,
02A7 CA1403	JZ	FILERD-3
02AA CDCF04	CALL	CRLF
02AD 117801	LXI	D, MSG2
02B0 CDED04	CALL	CRTMSG
02B3 CDCF04	CALL	CRLF
02B6 119101	LXI	D, MSG3
02B9 CDED04	CALL	CRTMSG
02BC 0E01	MVI	C, READC
02BE CD0500	CALL	BDOS
02C1 FE32	CPI	32H
02C3 CAD302	JZ	DBL
02C6 3E37	MVI	A, 55
02C8 320B01	STA	LNDEX
02CB 3E06	MVI	A, 6
02CD 320F01	STA	SKNDEX
02D0 C3DD02	JMP	BEGIN

DBL:

02D3 3E1C	MVI	A, 28
02D5 320B01	STA	LNDEX
02D8 3E03	MVI	A, 3
02DA 320F01	STA	SKNDEX

02DD AF	XRA	A		
02DE 320701	STA	LCOUNT		
02E1 320901	STA	PCOUNT		
02E4 320A01	STA	PCOUNT+1		
	; DETERMINE PRINT MODE - ALL OR PARTIAL			
02E7 CDCF04	CALL	CRLF		
02EA 111F02	LXI	D,	MSG13	
02ED CDED04	CALL	CRTMSG		
02F0 CDDA04	CALL	RDMSG		
02F3 FE50	CPI	'P'		
02F5 CC3505	CZ	PART		
	; READ FIRST RECORD			
02F8 3A6800	LDA	FCBRL		
02FB F5	PUSH	PSW		
02FC 3E24	MVI	A,	'\$'	
02FE 326800	STA	FCBRL		
0301 115D00	LXI	D,	FCBFN	
0304 211101	LXI	H,	TYTLE	
0307 1A	TITLOOP:			
	LDAX	D		
0308 77	MOV	M,	A	
0309 23	INX	H		
030A 13	INX	D		
030B FE24	CPI	'\$'		
030D C20703	JNZ	TITLOOP		
0310 F1	POP	PSW		
0311 326800	STA	FCBRL		
0314 11010A	LXI	D,	0A01H	
	FILRD:			
	PUSH	D		
0318 0E1A	MVI	C,	26	
031A CD0500	CALL	BDOS		
				; CHANGE DMA BUFFER ADDRESS

031D 115C00	LXI D,	FCB	
0320 0E14	MVI C,	READFR	
0322 CD0500	CALL BDOS		;READ FILE RECORD
0325 D1	POP D		
0326 F5	PUSH PSW		
0327 218000	LXI H,	80H	
032A 19	DAD D		
032B EB	XCHG		
032C F1	POP PSW		
032D FE00	CPI 0		;CHECK FOR ERRORS
032F CA1703	JZ FILERD		
0332 FE01	CPI 01		
0334 C4AA04	CNZ ERROR		;CHECK FOR END OF FILE
0337 3A0C01	LDA MODE		
033A FE2A	CPI '*'		
033C CA7105	JZ FIND		
033F 21000A	LXI H,	0A00H	
0342 C34903	JMP NEWPG		
REDY:			
0345 2A0301	LHLD LIMIT1		
0348 2b	DCX H		
	;ROUTINE STARTS NEW PAGE		
	NEWPG:		
0349 CDF203	CALL PLABEL		
	;ROUTINE BEGINS NEW LINE		
	NEWLN:		
034C 3E00	MVI A,	0	
034E 320801	STA CCOUNT		
0351 3A0C01	LDA MODE		
0354 FE2A	CPI '*'		;DETERMINE IF IN PARTIAL MODE
0356 CCC903	CZ CLEAN		
	GUTS:		
0359 CD7203	CALL GNB		;STARTS MAIN LOOP


```

035C FE0D      CPI      CR
035E CA9103    JZ      ENDLN
0361 FE09      CPI      TB
0363 CADC03    JZ      TAB
0366 CD7A03    CALL     PRCHAR
0369 C35903    JMP      GUTS
                ;END OF MAIN PROGRAM
*****
;SUBROUTINES
;BAD OPEN
BADF:          MVI      B,      01
                CALL     ERROR
                RET
                GNB:
0372 23        INX      H
0373 7E        MOV      A,      M
0374 FE1A      CPI      1AH
0376 CA0005    JZ      DONE
0379 C9        RET

;MAINTAINS CHARACTER COUNT
PRCHAR:        CALL     DRIVER
                LDA      CCOUNT
0380 3C        INR      A
0381 320601    STA      CCOUNT
0384 FE73      CPI      115
0386 C0        RNZ

                ; 115 CHARACTERS PER LINE

TRUNC:         CALL     GNB
                CPI      CR
0387 CD7203    JZ      ENDLN
038A FE0D
038C CA9103

```


038F 3E0D	MVI	A,	CR
;FINISHES LINE AND CHECKS LINE COUNT			
ENDLN:			
0391 CD7C04	CALL	DRIVER	
0394 CD7203	CALL	GNB	
0397 FE0A	CPI	LF	
0399 CA9E03	JZ	THERE	
039C 3E0A	MVI	A,	LF
THERE:			
039E CD7C04	CALL	DRIVER	
03A1 CD8704	CALL	BREAK	
03A4 3A1001	LDA	TEXT	
03A7 FE59	CPI	'Y'	
03A9 CA4C03	JZ	NEWLN	
03AC 3A0701	LDA	LCOUNT	
03AF 3C	INR	A	
03B0 320701	STA	LCOUNT	
03B3 E5	PUSH	H	
03B4 210B01	LXI	H,	LNDEX
03B7 BE	CMP	M	
03B8 E1	POP	H	
03B9 C24C03	JNZ	NEWLN	
;OUTPUT FORMFEED TO PRINTER; IF OUT OF PAPER CONDITION			
;EXISTS, RECEIPT OF FF TURNS PRINTER OFF. WHEN IN PARTIAL			
;PRINT MODE, THIS SPACES FIRST LINE TO ALIGN DESIRED FIRST			
;WORD IN PROPER COLUMN			
03E3 3E0C	MVI	A,	FF
03E5 CDA304	CALL	DRIVER	
03E8 3E00	MVI	A,	0
03EA 320701	STA	LCOUNT	
03C6 C34903	JMP	NEWPG	
CLEAN:			

03C9 3A0D01	LDA	NEAT	
03CC 47	MOV	B,	A
SWEEP:			
03CD 3E20	MVI	A,	20H
03CF CD7A03	CALL	PRCHAR	
03D2 05	DCR	B	
03D3 C2CD03	JNZ	SWEEP	0
03D6 3E00	MVI	A,	
03D8 320C01	STA	MODE	
03DB C9	RET		

;SKIPS SPACES TO NEXT TAB SETTING
TAB:

03DC 3A0801	LDA	CCOUNT	
03DF 47	MOV	B,	A
03E0 E6F8	ANI	0F8H	
03E2 C608	ADI	08H	
03E4 90	SUB	B	
03E5 47	MOV	B,	A

03E6 3E20	MVI	A,	20H
03E8 CD7A03	CALL	PRCHAR	
03EB 05	DCR	B	
03EC C2E603	JNZ	TBLOOP	
03EF C35903	JMP	GUTS	

; INCREMENTS PAGE NUMBER IN BCD

03F2 3A1001	LDA	TEXT	
03F5 FE59	CPI	'Y'	
03F7 C8	RZ		
03F8 E5	PUSH	H	
03F9 3A0F01	LDA	SKNDEX	

PGLOOP:

03FC 47	MOV	B,	A	
03FD 3E0A	MVI	A,	LF	
03FF CD7C04	CALL	DRIVER		
0402 05	DCR	B		
0403 C2FD03	JNZ	PGLOOP+1		
0406 115D01	LXI	D,	MSG1	
0409 CDF504	CALL	PRMSG		
040C 1600	MVI	D,	0	
040E 210901	LXI	H,	PCOUNT	
0411 7E	MOV	A,	M	
0412 3C	INR	A		
0413 27	DAA			
0414 77	MOV	M,	A	
0415 23	INX	H		
0416 7E	MOV	A,	M	
0417 CE00	ACI	0		
0419 77	MOV	M,	A	
041A E6F0	ANI	0FH		
041C 1F	RAR			
041D 1F	RAR			
041E 1F	RAR			
041F 1F	RAR			
0420 CD5104	CALL	PRPAGE	M	
0423 7E	MOV	A,		
0424 E60F	ANI	0FH		
0426 CD5104	CALL	PRPAGE		
0429 2B	DCX	H		
042A 7E	MOV	A,	M	
042B E6F0	ANI	0FH		
042D 1F	RAR			
042E 1F	RAR			
042F 1F	RAR			
0430 1F	RAR			
0431 CD5104	CALL	PRPAGE		
0434 7E	MOV	A,	M	
0435 E60F	ANI	0FH		

0437	CD5104	CALL	PRPAGE	30
043A	061E	MVI	B,	
LOOPER:				
043C	3E20	MVI	A,	20H
043E	CD7C04	CALL	DRIVER	
0441	05	DCR	B	
0442	C23C04	JNZ	LOOPER	
0445	211101	LXI	H,	TYTLE
0448	EB	XCHG		
0449	CD504	CALL	PRMSG	
044C	CD6404	CALL	PCR2LF	
044F	E1	POP	H	
0450	C9	RET		

;PRINTS PAGE NUMBER DIGIT
PRPAGE:

0451	C630	ADI	30H	
0453	FE30	CPI	30H	
0455	C25E04	JNZ	PRPG	
0458	47	MOV	B,	A
0459	7A	MOV	A,	D
045A	FE01	CPI	01	
045C	C0	RNZ		
045D	78	MOV	A,	B

PRPG:

045E	1601	MVI	D,	01
0460	CD7C04	CALL	DRIVER	
0463	C9	RET		

;PRINTER FORMAT CONTROL
PCR2LF:

0464	3E0D	MVI	A,	CR
0466	CD7C04	CALL	DRIVER	
0469	3E0A	MVI	A,	LF
046B	CD7C04	CALL	DRIVER	
046E	3E0A	MVI	A,	LF
0470	CD7C04	CALL	DRIVER	

0473 3A0701
0476 C603
0478 320701
047B C9

LDA LCOUNT
ADI 03
STA LCOUNT
RET

;CHECKS STATUS AND XMIT DATA TO USART
DRIVER:

047C F5

PUSH PSW

STS:

047D DB63
047F 0F
0480 D27D04
0483 F1
0484 D362
0486 C9

IN 63H
RRC
JNC STS
POP PSW
OUT 62H
RET

;CHECK BREAK KEY (ANY KEY) FOR INTERRUPT
BREAK:

0487 0E0B
0489 E5
048A CD0500
048D E1
048E 0F
048F D0
0490 E5
0491 CDCF04
0494 11F801
0497 CDED04
049A CDCF04
049D CDDA04
04A0 CDDA04
04A3 FE4B
04A5 E1
04A6 C0
04A7 C30005

MVI C, BRKF
PUSH H
CALL BDOS
POP H
RRC
RNC
PUSH H
CALL CRLF
LXI D, MSG10
CALL CRTMSG
CALL CRLF
CALL RDMSG
CALL RDMSG
CPI 'K'
POP H
RNZ
JMP DONE

;EMPTY UART BUFFER
;WAIT FOR NEXT CHAR


```
;PRINT ERROR MESSAGE ON CONSOLE
ERROR:
```

```
04AA E5
04AB CDCF04
04AE 3E07
04B0 CDE204
04B3 78
04B4 FE01
04B6 CABE04
04B9 FE03
04BB CAC704
```

```
PUSH H
CALL CRLF
MVI A, 07
CALL WRMSG
MOV A, B
CPI 01
JZ ERR1
CPI 03
JZ ERR3
```

```
ERR1:
```

```
04BE 11AB01
04C1 CDED04
04C4 C30005
```

```
LXI D, MSG4
CALL CRTMSG
JMP DONE
;FILE NOT FOUND
```

```
ERR3:
```

```
04C7 11BB01
04CA CDED04
04CD E1
04CE C9
```

```
LXI D, MSG7
CALL CRTMSG
POP H
RET
```

```
;CARRIAGE RETURN AND LINE FEED
CRLF:
```

```
04CF 3E0D
04D1 CDE204
04D4 3E0A
04D6 CDE204
04D9 C9
```

```
MVI A, CR
CALL WRMSG
MVI A, LF
CALL WRMSG
RET
```

```
;READ CHARACTER FROM CONSOLE
RDMSG:
```

```
04DA 0E01
```

```
MVI C, READC
```


04DC D5	PUSH D	
04DD CD0500	CALL BDOS	
04E0 D1	POP D	
04E1 C9	RET	
;WRITE CHARACTER TO CONSOLE		
	WRMSG:	
04E2 C5	PUSH B	
04E3 D5	PUSH D	
04E4 0E02	MVI C, TYPEC	
04E6 5F	MOV E, A	
04E7 CD0500	CALL BDOS	
04EA D1	POP D	
04EB C1	POP B	
04EC C9	RET	
;PRINTS MESSAGE ON CONSOLE		
	CRMSG:	
04ED 0E09	MVI C, 9	
04EF E5	PUSH H	
04F0 CD0500	CALL BDOS	
04F3 E1	POP H	
04F4 C9	RET	
;PRINTS MESSAGE ON PRINTER		
	PRMSG:	
04F5 1A	LDAX D	
04F6 FE24	CPI '4'	
04F8 C8	RZ	
04F9 CD7C04	CALL DRIVER	
04FC 13	INX D	
04FD C3F504	JMP PRMSG	
;SIGN OFF ON PRINTER		
	DONE:	
0500 CD6404	CALL PCR2LF	
0503 3A0B01	LDA LNDEX	

0506 D603	SUI	3H	
0508 2A0701	LHLD	LCOUNT	
050B BE	CMP	M	
050C FA1D05	JM	FINISH	
050F 3A1001	LDA	TEXT	
0512 FE59	CPI	'Y'	
0514 CA1D05	JZ	FINISH	
0517 11DF01	LXI	D,	MSG8
051A CDF504	CALL	PRMSG	
FINISH:			
051D CD6404	CALL	PCR2LF	
0520 3E0C	MVI	A,	FF
0522 CD7C04	CALL	DRIVER	
0525 3E50	MVI	A,	50H
0527 D363	OUT	063H	
0529 CDCF04	CALL	CRLF	
052C 11F201	LXI	D,	MSG9
052F CDED04	CALL	CRTMSG	
0532 C30000	JMP	0000H	

;SET UP TO PRINT PART OF PROGRAM
PART:

0535 CDCF04	CALL	CRLF	
0538 3E2A	MVI	A,	'*'
053A 320C01	STA	MODE	
053D 113D02	LXI	D,	MSG14
0540 CDED04	CALL	CRTMSG	
0543 CDCF04	CALL	CRLF	
0546 110009	LXI	D,	900H

;READ AND STORE STRING CHARACTERS-
STR1:

054A CDDA04	INX	D	
	CALL	RDMSG	
			;STRING1 BEGINS AT 901H
			;STRING2 BEGINS AT 911H


```

;DELIMITER IS CHARACTER 13H
;IF RUBOUT SELECTED, CORRECT IT

```

054D FE7F	CPI	7FH	
054F CAE405	JZ	UNDO1	
0552 12	STAX	D	
0553 FE2C	CPI	'	
0555 C24905	JNZ	STR1	
0558 3E13	MVI	A,	13H
055A 12	STAX	D	
055B 111009	LXI	D,	910H

STR2:

055E 13	INX	D	
055F CDDA04	CALL	RDMSG	
0562 FE7F	CPI	7FH	
0564 CAED05	JZ	UNDO2	
0567 12	STAX	D	
0568 FE0D	CPI	CR	
056A C25E05	JNZ	STR2	
056D 3E13	MVI	A,	13H
056F 12	STAX	D	
0570 C9	RET		

FIND:

0571 21010A	LXI	H,	0A01H	;FIND 1ST STRING AND APPEND ALL
0574 220301	SHLD	LIMIT1		
0577 2B	DCX	H		

RESET:

0578 110109	LXI	D,	901H	;AFTER TO TPA STARTING AT 0A01
057B 1A	LDAX	D		
057C FE13	CPI	13H		
057E CAB205	JZ	FIND28		

;LOCATE 1ST CHARACTER OF 1ST STRING

FIND1:

0581 23	INX	H
0582 BE	CMP	M

0583 C28105
0586 220301

JNZ FIND1
SHLD LIMIT1

;AFTER 1ST CHARACTER FOUND, CHECK ADDITIONAL CHARACTERS
;UNTIL STRING IS EXHAUSTED
NCR:

05A8 13
05A9 23
05EB 1A
05EC FE13
058E CA9E05
0591 BE
0592 C29805

INX D
INX H
LDAX D
CPI 13H
JZ FIND2
CMP M
JNZ FIND15

;IF NOT CORRECT STRING
;BEGIN SEARCH AGAIN

0595 C38905
0598 2A0301
059B C37805

JMP NCR
LHLD LIMIT1
JMP RESET

FIND15:

FIND2:

059E E5
059F 2A0301
05A2 5D
05A3 3E0A

PUSH H
LHLD LIMIT1
MOV E, L
MVI A, LF

;SET UP SPACING FOR 1ST LINE-
;DESIRE FIRST WORD TO PRINT IN
;PROPER COLUMN

FORMAT:

05A5 2B
05A6 BE
05A7 C2A505
05AA 7B
05AB 95
05AC D601
05AE 320D01
05B1 E1

DCX H
CMP M
JNZ FORMAT
MOV A, E
SUB L
SUI 1
STA NEAT
POP H

UNDO1:

05E4 1B	DCX	D
05E5 1A	LDAX	D
05E6 CDE204	CALL	WRMSG
05E9 1B	DCX	D
05EA C34905	JMP	STR1

UNDO2:

05ED 1B	DCX	D
05EE 1A	LDAX	D
05EF CDE204	CALL	WRMSG
05F2 1B	DCX	D
05F3 C35E05	JMP	STR2

BOARD:

THIS ROUTINE INITIALIZES THE 534 BOARD, THE TIMERS, AND THE TWO USARTS
NEEDED TO DRIVE THE IBM HIGH SPEED LINE AND THE MODEL 40 PRINTER

BASE ADDR OF 534 BOARD	60H
CMD ADDR OF LINE USART	61H
DATA ADDR OF LINE USART	62H
CMD ADDR OF PTR USART	63H
DATA ADDR OF PTR USART	62H

TWO MORE USARTS AND ONE 8255 PARALLEL INTERFACE AND THEIR TIMERS ARE
AVAILABLE ON THE 534 BOARD. NEW INTERFACES MUST BE PROGRAMMED BEFORE USE

05F6 F3	DI	;DISABLES 8080 INTERRUPTS
05F7 D36F	OUT	;RESETS BOARD
05F9 D36C	OUT	;SELECTS BOARD CONTROL BLOCK
05FB CD0306	CALL	;INITIALIZE PIT CHIPS


```

05FE CD1406      CALL    USART      ;INITIALIZE USARTS
0601 FB          EI              ;REENABLES INTERRUPTS
0602 C9          RET

;
;
;MUST SET UP TIMER CHIPS ACCORDING TO PAGE 3-12 OF 534 MANUAL
; CHIP 0 HAS THREE TIMERS ON IT
; TIMERS 0 AND 1 OF CHIP 0 ARE CONNECTED TO USARTS 1 AND 2
; RESPECTIVELY, DRIVING THE IBM LINE AND THE PRINTER
;

TIMER:
0603 D36C      OUT      6CH          ;SELECT CONTROL BLOCK
0605 3E76      MVI      A, 76H      ;
0607 D363      OUT      63H          ;SELECT TIMER 1 FOR PTR USART
0609 3E08      MVI      A, 8H      ;
060B D361      OUT      61H          ;SET N=8 IN TIMER 1
060D 3E00      MVI      A, 0H      ;CCLK/N=153.6KHZ FOR 9600 BAUD,
060F D361      OUT      61H          ;BRF=16X
0611 D36D      OUT      6DH          ;PUTS BOARD IN DATA BLOCK
0613 C9      RET

;
;
;SET UP BOTH USARTS WITH RESETS AND MODE WORDS
;

USART:
0614 3E5A      MVI      A, 5AH      ;1 STOP, PAR DISABLED, 7 BITS
0616 D363      OUT      63H
0618 3E33      MVI      A, 33H
061A D363      OUT      63H
061C C9      RET

```

APPENDIX J

G02 ASSEMBLY PROGRAM

1 AUG 1978

```

**
**MDS 8080 PROGRAM INTERFACES DATEL ST-800 ANALOG
**TO DIGITAL CONVERTER BOARD AND INTEL DYNAMIC
**MEMORY ACCESS CONTROLLER FOR HIGH SPEED DATA
**ACQUISITION ---
**MAXIMUM OF 16 CHANNELS ARE INPUT, CONVERTED,
**AND STORED IN MEMORY AT A RATE OF 40 KHZ ---
**PROGRAMMABLE INTERRUPT CONTROLLER AND
**INTERVAL TIMERS ON THE INTEL SBC 534 BOARD ARE
**INTERFACED TO PROVIDE VARIABLE SCAN RATES
**OF ONE TO 2000 SCANS PER SECOND *****
*****
ORG 100H
0100 C39C05          JMP      START
                      ;
                      ;EQUATES
                      ;
CR      EQU          0DH          ;CARRIAGE RETURN
LF      EQU          0AH          ;LINE FEED
BDOS    EQU          5H          ;BDOS ENTRY POINT
DMACMD  EQU          17H          ;DMA COMMAND WORD
REVRT   EQU          20H          ;CPU INTERRUPT CLEAR COMMAND
R04     EQU          20H          ;RESTART 04 ADDRESS
R05     EQU          28H          ;RESTART 05 ADDRESS
DMA     EQU          40H          ;DMA BASE ADDRESS
SBC     EQU          60H          ;SBC 534 BASE ADDRESS
JUMP    EQU          0C3H          ;JUMP INSTRUCTION
MASK    EQU          0FCH          ;MASK ALTERATION PORT

```



```

MEMORY EQU      ;DATA MEMORY BUFFER ADDRESS
;
;
;DATA SAVES
;
WCNT: DS        WORD LENGTH SETTING ( X 2 )
ACHAN: DS        START CHANNEL
BCCHAN: DS        FINAL CHANNEL
INTVL4: DS        TIMER 4 SETTING
INTVL5: DS        TIMER 5 SETTING
RRCOUNT: DS        1H ;SCAN RATE REGISTER
PCOUNT: DS        1H ;# DATA POINTS REGISTER
LIMIT: DS        1H ;MSB OF UPPER MEMORY LIMIT
FLNAME: DB        0,'DATA1',0,0,0,0
DS        17D
STACK: DS        20H ;SAVE ROOM FOR STACK
SSTKBTM EQU      $ ;INITIATE STACK POINTER HERE
;
;
;MESSAGES
;
MSG1: DB        CR,LF,'ENTER STARTING CHANNEL $'
MSG2: DB        CR,LF,'ENTER FINAL CHANNEL $'
MSG3: DB        CR,LF,'CARRIAGE RETURN TO BEGIN $'
MSG4: DB        CR,LF,'TRY AGAIN, TURKEY $'
MSG5: DB        CR,LF,'ENTER DESIRED NUMBER OF DATA POINTS'
DB        CR,LF,LF,'ENTER DATA POINTS'
DB        CR,LF,LF,'A 1024'
DB        CR,LF,LF,'B 4096'
DB        CR,LF,LF,'C 5120'
DB        CR,LF,LF,'D 10240'
DB        CR,LF,LF,'E 26112'
DB        CR,LF,LF,'$'
MSG6: DB        CR,LF,'SELECT SCAN RATE',CR,LF,LF
DB        CR,LF,'ENTER SCANS/SEC'
DB        CR,LF,'MAX CHANNELS',CR,LF

```



```

DB      A          5000      1',CR,LF
DB      B          4800      2',CR,LF
DB      C          2600      8',CR,LF
DB      D          2200     12',CR,LF
DB      E          1800     16',CR,LF
DB      F          1000     16',CR,LF
DB      G          100      16',CR,LF
DB      H           10       16',CR,LF
DB      I            1       16',CR,LF

MSG7:   DB      CR,LF,'WRITE DATA FILE ON DISK?? (Y/N) $'
MSG8:   DB      CR,LF,'ANOTHER DATA RUN DESIRED?? (Y/N) $'
MSG9:   DB      CR,LF,'DISK FULL - TRY ANOTHER - RETURN WHEN READY $'
MSG10:  DB      CR,LF,'DISK WRITE ERROR - TRY ANOTHER - RETURN WHEN READY $'
;
;
;
;
; START:
059C    314E01      LXI      SP,          ;SET UP STACK POINTER
059F    3EC3        MVI      A,          ;JUMP INSTRUCTION
05A1    322000      STA      R04         ;SET UP INTERRUPT
05A4    322800      STA      R05         ;JUMP VECTORS
05A7    21D906      LXI      H,          ;ADDR OF INT 4 ROUTINE
05AA    222100      SHLD     R04+1
05AD    211B07      LXI      H,          ;ADDR OF INT 5 ROUTINE
05B0    222900      SHLD     R05+1

; CHANGE CPU MASK TO ACCEPT RST 04 AND RST 05 INTERRUPTS
;
05B3    3E4E        MVI      A,          4EH      ;ALLOWS RST 0,4,5,7
05B5    D3FC        OUT      MASK

; GET VALUES FOR INITIAL AND FINAL CHANNELS AND WORD LENGTH
; SETUP:

```



```

05F7 CDA907 CALL OOPS ;NOTHING ELSE IS VALID
05FA C3DE05 JMP POINT

; APOINT: MVI A, 0EH
JMP DOWN

BPOINT: MVI A, 1AH
JMP DOWN

CPOINT: MVI A, 32H
JMP DOWN

DPOINT: MVI A, 5AH
JMP DOWN

EPOINT: MVI A, 0D8H
JMP DOWN

DOWN: STA LIMIT

;
; LIMIT IS NOW SET UP
; NEXT DETERMINE DESIRED SCAN RATE
;
; RATE:
;

0616 11E602 LXI D, MSG6
0619 0E09 MVI C, 9H
061B CD0500 CALL BDOS
061E CDA307 CALL KEY
0621 110100 LXI D, RCOUNT
0624 320A01 STA 'A'
0627 FE41 CPI 'A'
0629 CA5A06 JZ ARATE
062C FE42 CPI 'B'
062E CA6006 JZ BRATE
0631 FE43 CPI 'C'

; PROMPT USER
; GET USER'S CHOICE OF RATES
; LOAD D FOR LATER USE
; SAVE FOR FUTURE USE
; SEE IF A ENTERED
; SEE IF B ENTERED
; SEE IF C ENTERED

```


JZ	0633	CA6606	CRATE	
CPI	0636	FE44	'D	;SEE IF D ENTERED
JZ	0638	CA6C06	DRATE	
CPI	063B	FE45	'E	;SEE IF E ENTERED
JZ	063D	CA7206	ERATE	
CPI	0640	FE46	'F	;SEE IF F ENTERED
JZ	0642	CA7806	FRATE	
CPI	0645	FE47	'G	;SEE IF G ENTERED
JZ	0647	CA7E06	GRATE	
CPI	064A	FE48	'H	;SEE IF H ENTERED
JZ	064C	CA8406	HRATE	
CPI	064F	FE49	'I	;SEE IF I ENTERED
JZ	0651	CA8D06	IRATE	
CALL	0654	CDA907	OOPS	;MUST BE A - I TO BE VALID
JMP	0657	C31606	RATE	

; ; SET UP REGISTERS ACCORDINGLY

```

; ARATE:      LXI      217A00      ; COUNTS ARE DETERMINED IN
;              JMP      C39306      ; THE FOLLOWING MANNER:
;
; PRATE:      LXI      218700      ;
;              JMP      C39306      ; N = 1.2288 MHZ
;
; CRATE:      LXI      213E01      ;
;              JMP      C39306      ; SCANS/SEC
;
; DRATE:      LXI      21B601      ; SINCE THE SERVICE ROUTINE
;              JMP      C39306      ; REQUIRES 98.5 MICROSECONDS
;
; ERATE:      LXI      213202      ; TO COMPLETE, DETERMINE THE
;              JMP      C39306      ; SETTING BEING ENTERED HERE
;
; FRATE:      LXI      215404      ; BY THE FORMULA
;              JMP      C39306      ;
;
;              LXI      215404      ; N* = N - 1.2288 X 98.5
;              JMP      C39306      ;

```



```

067E 21872F      LXI      H,      2F87H      ;
0681 C39306      JMP      RASET      ;
;
0684 21C4EF      LXI      H,      0EFC4H      ; NORMALLY THE SETTING IN TIMER 5
0687 110200      LXI      D,      2H          ; IS 1, BUT FOR INTERVALS OVER
068A C39306      JMP      RASET      ; 50 MILLISECONDS, THE COUNT N
;
068D 21FAEF      LXI      H,      0EFAH       ; MUST BE DIVIDED BY SOME
0690 111400      LXI      D,      14H        ; NUMBER M TO REDUCE THE COUNT
;
0693 220601      SHLD     INTVL4      ; BELOW 0FFFFH; THEN THE TIMER 5
0696 EB          XCHG              ; SETTING INCREASED ACCORDINGLY
0697 220801      SHLD     INTVL5      ;
;
; RATE OF SCAN IS NOW SET INTO EFFECT
;
; BEGIN:
;
; NOW READY TO BEGIN SCANNING WHEN PROMPTED
069A 117F01      LXI      D,      MSG3
069D 0E09        MVI      C,      9H
069F CD0500      CALL     BDOS
06A2 CDA307      CALL     KEY
;
;
;
06A5 110000      LXI      D,      0H          ; ZERO DE REG
06A8 3A0301      LDA      WCNT
06AB 5F          MOV      E,      A          ; VALUE OF LENGTH REG
;
; SET UP DMA AND ST-800 BOARDS
;
06AC CDBE06      CALL     DMASET
06AF CDED06      CALL     TIMSET
;

```



```

006B6 D349
006B8 3A0301
006BB D34C
006BD AF
006BE D34D
006C0 21000A
006C3 7D
006C4 D34E
006C6 7C
006C7 D34F
006C9 3A0401
006CC D340
006CE 3A0501
006D1 D341
006D3 3E17
006D5 D34A
006D7 FB
006D8 C9

;
;
; DMASET:
      OUT DMA+9H
      LDA WCNT
      OUT DMA+0CH
      XRA A
      OUT DMA+0DH
      LXI H, MEMORY
      MOV A, L
      OUT DMA+0EH
      MOV A, H
      OUT DMA+0FH
      LDA ACHAN
      OUT DMA
      LDA BCHAN
      OUT DMA+1H
      MVI A, DMACMD
      OUT DMA+0AH
      EI
      RET

;
; DMA NOW READY TO GO WHEN COMMAND WORD IS ISSUED
;
; RESET4:
      OUT DMA+9H
      LDA WCNT
      OUT DMA+0CH
      XRA A
      OUT DMA+0DH
      MVI A, REVRT
      OUT 0FDH
      MVI A, DMACMD
      OUT DMA+0AH
      EI
      RET

      ;RESET DMA
      ;LENGTH REG SETTING
      ;MSB OF LENGTH REG IS 0
      ;CLEARS INT 4 FROM CPU
      ;INTERRUPT PENDING STACK
      ;COMMAND BYTE
      ;REENABLES INTERRUPTS
      ;DMA IS READY TO GO

```



```

0705 3A0901 LDA INTVL5+1 ;MSB OF TIMER 5 COUNT
0708 D366 OUT SBC+6H
070A 3E76 MVI A, 76H ;SELECT TIMER 4 AS CLOCK
070C D367 OUT SBC+7H ;FOR TIMER 5
070E 3A0601 LDA INTVL4 ;LSB OF TIMER 4 COUNT
0711 D365 OUT SBC+5H
0713 3A0701 LDA INTVL4+1 ;MSB OF TIMER 4 COUNT
0716 D365 OUT SBC+05H

; INTERRUPT TIMER IS NOW SET AND RUNNING
;
; OUT DMA+2H ;DMA "GO" INSTRUCTION
;
; DMA IS NOW SET AND RUNNING
;
; RET
;
;
; ROUTINE TO SERVICE INTERRUPT 5 FROM INTERRUPT TIMER
;
; RESET5: MVI A, 76H ;STOPS TIMER 4
; OUT SBC+7H
; LDA INTVL5 ;RESET LSB OF TIMER 5
; OUT SBC+6H ;(REMOVES INT 4 FROM BUS)
; LDA INTVL5+1 ;RESET MSB OF TIMER 5
; OUT SBC+6H
; MVI A, REVRT ;RESETS CPU
; OUT 0FDH ;REENABLES INTERRUPTS
; EI
;
; NEED TO KEEP TRACK OF MEMORY AREA USED TO PREVENT OVER
; RUNNING LIMIT
;
; DAD D ;DE REG CONTAINS WORDLENGTH
072E 19

```



```

072F 3A0C01      LDA    LIMIT
0732 BC          CMP    H
0733 CAB207      JZ     DONE          ;EXIT PROGRAM

;
; IF MEMORY SPACE OKAY, RESET TIMER AND CONTINUE
;
0736 3A0601      LDA    INTVL4          ;RESET LSB OF TIMER 4
0739 D365        OUT    SBC+5H
073B 3A0701      LDA    INTVL4+1        ;RESET MSB OF TIMER 4
073E D365        OUT    SBC+5H

;
; INTERRUPT TIMERS RUNNING AGAIN
;
0740 D342        OUT    DMA+2H

;
; DMA RUNNING AGAIN
;
0742 C9          RET

;
;
;
; ROUTINE TO READ IN INITIAL AND FINAL CHANNELS
;
DIGIT1:  LXI    D,    MSG1
          MVI    C,    9
          CALL   BDOS
          CALL   KEY
          CPI    CR
          JZ     DIGIT1
          SUI    30H
          STA    ACHAN
          CALL   KEY
          CPI    CR
          JZ     DIGIT2

0743 114E01      LXI    D,    MSG1
0746 0E09        MVI    C,    9
0748 CD0500      CALL   BDOS
074B CDA307      CALL   KEY
074E FE0D        CPI    CR
0750 CA4307      JZ     DIGIT1
0753 D630        SUI    30H
0755 320401      STA    ACHAN
0758 CDA307      CALL   KEY
075B FE0D        CPI    CR
075D CA7507      JZ     DIGIT2

; PROMPT USER
; GET ENTERED CHARACTER
; REDUCE ASCII
; SEE IF SECOND CHAR

```



```

07A3 0E01      MVI      C,      1H
07A5 CD0500    CALL      BDOS
07A8 C9        RET

;
;
; ROUTINE PRINTS MESSAGE IF TOO MANY CHARACTERS
;
OOPS:
07A9 119B01    LXI      D,      MSG4
07AC 0E09      MVI      C,      9
07AE CD0500    CALL      BDOS
07B1 C9        RET

;
;
; DONE:
07B2 F1        POP      PSW
07B3 11F104    LXI      D,      MSG7
07B6 0E09      MVI      C,      9H
07B8 CD0500    CALL      BDOS
07BB CDA307    CALL      KEY
07BE FE4E      CPI      'N'
07C0 CAC907    JZ      GETMOR
U07C3 CD0000   CALL      CRLF
07C6 C3E607    JMP      FLFILE

;
;
; GETMOR:
07C9 111405    LXI      D,      MSG8
07CC 0E09      MVI      C,      9H
07CE CD0500    CALL      BDOS
07D1 CDA307    CALL      KEY
07D4 FE59      CPI      'Y'
07D6 CADC07    JZ      RERUN

;
; OTHERWISE, ITS TIME TO QUIT
;
;
; DUMMY POP
; SEE IF USER WANTS
; FILE WRITTEN
; CHECK ANSWER
; IF NO, CONTINUE
; IF YES, GO WRITE
; SEE IF USER WANTS
; ANOTHER RUN
; CHECK ANSWER
; IF YES, GO BACK

```



```

07D9 C30000      EXIT:      JMP      0H      ;WARM BOOT
;
;SET UP FOR ANOTHER RUN
;
RERUN:
07DC 3A1201      LDA      FLNAME+5      ;INCREMENT FILE NAME
07DF 3C          INR      A
07E0 321201      STA      FLNAME+5
07E3 C39A06      JMP      BEGIN
;
;NEXT ROUTINE CREATES AND WRITES A DISK FILE -
;THE FIRST FILE RECORD CONTAINS INFORMATION
;WHICH WILL FACILITATE LATER RETRIEVAL OF THE
;DATA ---
;THE FIRST FILE RECORD CONTAINS THE DATA FILE
;NAME, FIRST CHANNEL, FINAL CHANNEL, SCAN RATE
;CODE LETTER, AND DATA POINTS CODE LETTER ---
;THE REMAINDER OF THE FIRST FILE RECORD IS ZEROES
;
;FILE:
;CREATE FILE ON DISK
;
07E6 0E13      MVI      C,      19      ;DELETE OLD FILE, SAME NAME
07E8 110D31      LXI      D,      FLNAME
07EB CD0500      CALL     BDOS
07EE 0E16      MVI      C,      22
07F0 110D01      LXI      D,      FLNAME
07F3 CD0500      CALL     BDOS
07F6 FEFF      CPI      255
07F8 CA6908      JZ      NOROOM
07FB AF          XRA      A
07FC 322D01      STA      FLNAME+32
;

```



```

;NEXT SET UP FIRST FILE RECORD
;
RECORD:
07FF 3E00      MVI A, 0H
0801 118009    LXI D, MEMORY-80H
0804 0680      MVI B, 80H
;ZERO OUT RECORD

RLOOP:
0806 12        STAX D
0807 13        INX D
0808 05        DCR B
0809 C20608    JNZ RLOOP

;
;FILL IN FILE RECORD DATA
;
RLOOP2:
080C 010E01    LXI B, FLNAME+1
080F 118009    LXI D, MEMORY-80H
0812 2605      MVI H, 5H
;COPY FIRST 5 LETTERS
;OF FILE NAME INTO
;RECORD

RLOOP2:
0814 0A        LDAX B
0815 12        STAX D
0816 03        INX B
0817 13        INX D
0818 25        DCR H
0819 C21408    JNZ RLOOP2

;
081C 3A0401    LDA ACHAN
081F 12        STAX D
0820 13        INX D
0821 3A0501    LDA BCHAN
0824 12        STAX D
0825 13        INX D
0826 3A0A01    LDA RCOUNT
0829 12        STAX D
082A 13        INX D
082B 3A0B01    LDA PCOUNT
082E 12        STAX D
;FIRST CHANNEL
;FINAL CHANNEL
;SCAN RATE CODE
;DATA POINT CODE

```



```

;
;FIRST FILE RECORD NOW CONTAINS APPROPRIATE INFORMATION
;
;
;SINCE DMA PUT PAIRS OF DATA BYTES INTO MEMORY IN REVERSE
;ORDER, WANT TO REVERSE THEM BEFORE WRITING ON DISK
;
;FLIP:
082F 3A2C01 LDA LIMIT ;UPPER LIMIT ON MEMORY USED
0832 21000A LXI H, MEMORY ;BEGINNING OF DATA
;
;FLOP:
0835 46 MOV B, ;GET LSB
0836 23 INX H
0837 4E MOV C, ;GET MSB
0838 70 MOV M, B ;PUT LSB
0839 2B DCX H
083A 71 MOV M, C ;PUT MSB
083B 23 INX H
083C 23 INX H
083D BC CMP H ;CHECK AGAINST LIMIT
083E C23508 JNZ FLOP
;
;DATA PAIRS NOW IN CORRECT ORDER
;
;
;READY TO START WRITING ONTO DISK
;
;FWRITE:
;
;
;LXI D, MEMORY-80H ;INFO RECORD
;
;FLOOP:
0841 118009 LXI D, ;SAVE POINTER
0844 D5 PUSH D ;
0845 0E1A MVI C, 26 ;CHANGE BUFFER ADDRESS
0847 CD0500 CALL BDOS

```


084A 110D01	LXI D,	FLNAME	
084D 0E15	MVI C,	21	
084F CD0500	CALL BDOS		;WRITE ONE RECORD
0852 D1	POP D		;RETRIEVE POINTER
0853 F5	PUSH PSW		;WILL CHECK LATER
0854 218000	LXI H,	80H	
0857 19	DAD D		;INCREMENT POINTER
0858 EB	XCHG		;BY 80H
0859 F1	POP PSW		
085A FE00	CPI 0H		;CHECK FOR WRITE ERRORS
085C C27708	JNZ ERROR		
085F 3A0C01	LDA LIMIT		;CHECK END OF DATA
0862 BA	CMP D		;MSB ONLY
0863 CA8508	JZ CLOSE		
0866 C34408	JMP FLOOP		;GO DO ANOTHER RECORD

			; THIS CONTINUES UNTIL ALL DATA WRITTEN ONTO DISK
			; ROUTINE INFORMS USER THAT DISK OR DIRECTORY IS FULL
			; ; ; ; ;
			NOROOM:

0869 113805	LXI D,	MSG9	
086C 0E09	MVI C,	9H	
086E CD0500	CALL BDOS		
0871 CDA307	CALL KEY		;WAIT FOR RESPONSE
0874 C3E607	JMP FLFILE		;TRY ANOTHER WRITE

			; ; ;
			ERROR:

0877 116705	LXI D,	MSG10	
087A 0E09	MVI C,	9H	
087C CD0500	CALL BDOS		;INFO USER OF ERROR
087F CDA307	CALL KEY		;CHECK FOR RESPONSE
0882 C3E607	JMP FLFILE		; ;


```

0885 110D01
0888 0E10
088A CD0500
088D C3C907

; IF ERROR OCCURRED IN WRITING ON DISK, ANOTHER WRITE SHOULD
; BE ATTEMPTED ON ANOTHER DISK
;
; WHENEVER DATA WRITE IS COMPLETED, NEED TO CLOSE FILE
;
CLOSE:
    LXI    D,    FINAME
    MVI    C,    16
    CALL   BDOS
    JMP    GETMOR
;CHECK WITH USER

;*****
END 100H
0890

```


APPENDIX K

PATCH FOR CP/M BIOS PROGRAM

```

;PATCH TO CP/M BIOS PROGRAM
;
;ALTERS JUMP VECTOR BY READDRESSING JUMPS TO
;THE LIST OUT (LO) DEVICE.
;JUMP VECTOR INSTEAD POINTS TO ALTERNATE ROUTINE
;WHICH SENDS CHARACTER TO MODEL 40 PRINTER.
;PRINTER MUST HAVE BEEN PREVIOUSLY SET UP
;BY AN INDEPENDENT ROUTINE (ON.COM)
;

```

BE00	C344BE	JMP	BOOT
BE03	C354BE	JMP	WBOOT
BE06	C3F2BE	JMP	CONST
BE09	C3F5BE	JMP	CONIN
BE0C	C3FBBE	JMP	CONOUT
BE0F	C3E7BF	JMP	PATCH
BE12	C301BF	JMP	PUNCH
BE15	C304BF	JMP	READER
BE18	C307BF	JMP	HOME
BE1B	C30CBF	JMP	SELDSK
BFE7		ORG	0BFE7H

```

PATCH:
IN      63H      ;CHECK USART STATUS
ANI     1
JZ      PATCH
MOV     A,C
OUT     62H
RET
;PUT BYTE IN ACCUM
;SEND TO USART

```


APPENDIX L

ON ASSEMBLY PROGRAM

;THIS ROUTINE INITIALIZES THE INTEL SBC 534 BOARD,
;THE TIMER, AND THE USART NEEDED TO DRIVE THE
;MODEL 40 PRINTER
;

0100

ORG 100H

;BASE ADDR OF 534 BOARD 60H
;CMD ADDR OF PRINTER USART 63H
;DATA ADDR OF PRINTER USART 62H
;

START:

0100 310002
0103 D36F
0105 D36C

LXI SP, 200H ;SET UP STACK
OUT 6FH ;RESETS 534 BOARD
OUT 6CH ;SELECTS CONTROL BLOCK

TIMER:

0107 3E76
0109 D363
010B 3E08
010D D361
010F 3E00
0111 D361

MVI A, 76H ;SELECT TIMER 1 FOR
OUT 63H ;PRINTER USART --
MVI A, 8H ;SET N=8 IN TIMER 1
OUT 61H ;CCLK/N = 153.6KHZ FOR 9600
MVI A, 0H ;BAUD, BRF = 16X
OUT 61H

USART:

0113 D36D

OUT 6DH ;SELECT DATA BLOCK
;MODE WORD - SETS UP 1 STOP BIT, ODD PARITY
;ENABLED, 7 BIT WORD, AND A BAUD RATE
;FACTOR OF 16X


```

;
0115 3E5A      MVI    A,    5AH      ;MODE WORD
0117 D363      OUT     63H      ;COMMAND PORT

;COMMAND WORD - SETS RTS, ERROR RESET, DTR,
;AND XMIT ENABLE
;

0119 3E33      MVI    A,    33H      ;COMMAND WORD
011B D363      OUT     63H      ;COMMAND PORT

011D C30000     JMP     0H        ;SOFT BOOT

0120           END     100H

```


REDUCE FORTRAN PROGRAM

APPENDIX M

```

C ** FOURIER COEFFICIENT DETERMINATION **
C * PROGRAM INPUT CONSISTS OF CHANNELS "J1" TO "JMAX" OF
C DISCRETIZED DATA USING A COMMON TIME BASE FOR THE SAMPLINGS.
C * PROGRAM OUTPUT CONSISTS OF FOURIER COEFFICIENTS FOR THE
C VARIOUS CHANNELS, INCLUDING OPTIONS FOR HIGHER HARMONICS.
C RELATIVE PHASING BETWEEN THE CHANNELS IS OBTAINED.
C
      DIMENSION Y(5),RMS(5),A(5,5),B(5,5),C(5,5),PHI(5,5),IX(5,500)
      1,X(5,500)
      1 FORMAT (1H0,'ENTER DISK FILE NUMBER (12)'/)
      2 FORMAT (12)
      3 FORMAT (1H0,'ENTER FILE NO. (12), NUMBER OF CHANNELS (12), SCAN')
      4 FORMAT (1H,'RATE (15), FUNDAMENTAL FREQUENCY (F6.0), NUMBER')
      5 FORMAT (1H,'OF DATA POINTS (15), COORDINATION NUMBER (18)'/)
      6 FORMAT (2I2,15,F6.0,15,I8)
      7 FORMAT (1H0)
      8 FORMAT (4Z4)
      9 FORMAT (5X,I4,4(5X,F8.5))
      11 FORMAT (1H1,'DATA',I2,/)
      12 FORMAT (1H,'15, DATA POINTS,')
      13 FORMAT (1H,'SCAN RATE',I5,'HERTZ,')
      14 FORMAT (1H,'COORDINATION NUMBER',I8,/)
      J1 = 1
      JMAX = 1
      IDISK = 1
      IRATE = 1
      ICOORD = 000
      F1 = 1.
      IFNAME = 1
      PI = 3.141592654
      WRITE (6,1)

```



```

C      READ (5,2) IDISK
C      WRITE (6,3)
C      WRITE (6,4)
C      WRITE (6,5)
C      READ (5,6) IFNAME,JMAX,IRATE,F1,IR,ICOORD
C
C      ** TRUNCATE DATA SET TO INTEGER NO. OF FUNDAMENTAL PERIODS **
C      IR = NO. OF DATA RECORDS (OPTION SELECTABLE)
C      J1 = INITIAL DATA CHANNEL IDENT.
C      JMAX = FINAL DATA CHANNEL IDENT. (JMAX .GE.1 AND .LE.16)
C      F1 = FUNDAMENTAL FREQUENCY (HZ)
C      DELT = SAMPLE TIME FOR A DATA CHANNEL (SEC)
C      ICOORD= COORDINATION NO.
C      IP = INTEGER NO. OF FUNDAMENTAL PERIODS
C      M = INTEGER NO. SAMPLES FOR EACH CHANNEL (TRUNCATED FORM)
C      N = IR/JMAX
C      AN = N
C      RATE = IRATE
C      DELT = 1./RATE
C      IP = IFIX(AN*F1*DELT)
C      AP = IP
C      M = IFIX(AP/(F1*DELT))
C
C      NEXT READ IN SAMPLED DATA FROM DISK FILE
C
C      DO 30 I = 1,M
C          READ (IDISK,8) (IX(J,I), J = J1,JMAX)
C      30 CONTINUE
C
C      SCALE INTEGER DATA AND CONVERT TO REAL NUMBERS
C
C      DO 40 I = 1,M
C          DO 35 J = J1,JMAX
C              IF (IX(J,I).GT.2047) GO TO 32
C              AAA = IX(J,I)
C              GO TO 33

```



```

32 AAA = IX(J,I) - 65536
33 CONST = 5./2047.
   X(J,I) = CONST * AAA
35 CONTINUE
40 CONTINUE

C
C
C
C
ECHO SCALED DATA VALUES TO CONSOLE

WRITE (6,7)
WRITE (6,11) IFNAME
WRITE (6,12) IR
WRITE (6,13) IRATE
WRITE (6,14) ICOORD
DO 45 I = 1,20
   WRITE (6,9) (I,(X(J,I), J=J1,JMAX))
45 CONTINUE

C
C
C
C
** FIND CHANNEL BIAS AND R.M.S. **
   Y(J) = AVE. VALUE OF CHANNEL "J"
   RMS(J) = RMS VALUE OF CHANNEL "J"
** REMOVE BIAS FROM DATA **
50 DO 59 J=J1,JMAX
   AVE = 0.0
51 DO 52 I=1,M
   AVE = AVE + X(J,I)
52 CONTINUE
   AM = M
   Y(J) = (1./AM)*AVE
   X2 = 0.0
53 DO 54 I=1,M
   X(J,I) = X(J,I) - Y(J)
   X2 = X2 + X(J,I)**2
54 CONTINUE
   X2 = (1./AM)*X2

```



```

RMS(J) = SQRT(X2)
59 CONTINUE
65 WRITE(6,1000) J1,JMAX,ICoord
  WRITE(6,1001) IR,DELT,F1
  WRITE(6,1002) M,N
  WRITE(6,1003)
70 DO 71 I=J1,JMAX
  WRITE(6,1010) I,Y(I),RMS(I)
71 CONTINUE
C ** FOURIER COEFFICIENT EVALUATION BRANCH **
C   KMAX = MAX. HARMONIC DESIRED
C   DELTAU = INTERCHANNEL SAMPLE DELAY (SEC)
C   X(J,I) = DATA ARRAYS (D.C. BIAS REMOVED)
C   J = DATA CHANNEL, J1 TO JMAX
C   I = DISCRETIZED SAMPLE INDEX, I=1 TO M
100 DELTAU = 0.
  KMAX = 2
110 DO 123 K=1,KMAX
  AK = K
  ARG = 2.*PI*F1*AK*DELT
  S1 = SIN(ARG)
  C1 = COS(ARG)
115 DO 122 I=J1,JMAX
  AI = (I-1)
  ARG = 2.*PI*F1*AK*(DELT + (AI*DELT))
  S2 = SIN(ARG)
  C2 = COS(ARG)
  A(K,I)=0.0
  B(K,I)=0.0
120 DO 121 L=1,M
  A(K,I)= A(K,I) + X(I,L)*C2
  B(K,I)= B(K,I) + X(I,L)*S2
  AC2 = C2*C1 - S2*S1
  AS2 = S2*C1 + C2*S1
  C2 = AC2
  S2 = AS2

```



```

121 CONTINUE
   AM = M
   A(K,I) = (2./AM)*A(K,I)
   B(K,I) = (2./AM)*B(K,I)
   C(K,I) = Sqrt(A(K,I)**2 + B(K,I)**2)
   A1 = ABS(A(K,I))
   B1 = ABS(B(K,I))
   IF(A1.LT.0.001.AND.B1.LT.0.001) GO TO 200
   PHI(K,I) = ATAN2(-B(K,I),A(K,I))*(180./PI)
   GO TO 125
200 PHI(K,I) = 0.0
125 CONTINUE
122 CONTINUE
123 CONTINUE
130 DO 137 K=1,KMAX
   WRITE(6,1020) K
135 DO 136 I=J1,JMAX
   WRITE(6,1025) I,A(K,I),B(K,I),PHI(K,I),C(K,I)
136 CONTINUE
137 CONTINUE
1000 FORMAT (1H1,4X,16HINITIAL CHANNEL:,T25,I2/7X,14HFINAL CHANNEL:,
1 T25,I2/ 7X,14HCOORD. NUMBER:,T25,I8,/)
1001 FORMAT(3X,18HTOTAL NO. SAMPLES:,T25,I5/
1 2X,19HSCAN PERIOD (SEC.):,T25,E11.4/1X,20HREFERENCE FREQ (HZ):,
2 T25,E11.4//)
1002 FORMAT(1X,20HDATA PTS./CH., USED:,,T25,I4,T35,7HAVAIL.:,T45,I4//)
1003 FORMAT(5X,'SIGNAL BIAS AND R.M.S. VALUES',/
1 2X,'CHANNEL',T15,'BIAS',T23,'R.M.S.'//)
1010 FORMAT(4X,I2,T12,F7.4,T22,F7.4)
1020 FORMAT(1H0,4X,'FOURIER COEFFICIENTS FOR HARMONIC',I3/
1 2X,'CHANNEL',T14,'COS',T24,'SIN',T34,'PHASE',T44,'MAG')
1025 FORMAT(4X,I2,T12,F7.4,T22,F7.4,T32,F7.2,T42,F7.4)
500 FORMAT (1H0,2X,'INDEX',T13,'X(1,I)',T23,'X(2,I)' /)
501 FORMAT (4X,I3,T12,F7.4,T22,F7.4)
      STOP
      END

```


APPENDIX N

DATA 3

1024 DATA POINTS

SCAN RATE 300 HERTZ

COORDINATION NUMBER 911001

1	1.10650	1.09673	1.09428	1.08207
2	-0.05862	-0.06839	-0.07328	-0.08305
3	-1.21641	-1.22374	-1.23351	-1.23840
4	-2.14704	-2.15437	-2.15681	-2.16170
5	-2.69419	-2.69419	-2.69663	-2.69663
6	-2.76991	-2.76746	-2.76746	-2.76746
7	-2.36688	-2.36444	-2.35955	-2.35466
8	-1.53639	-1.52907	-1.52418	-1.51685
9	-0.42013	-0.41280	-0.40547	-0.39814
10	0.75721	0.76453	0.77186	0.77919
11	1.82218	1.82706	1.83195	1.83928
12	2.56961	2.57206	2.57694	2.57938
13	2.86517	2.86517	2.86517	2.86517
14	2.69907	2.69663	2.69419	2.69174
15	2.06400	2.05911	2.05178	2.04690
16	1.07474	1.06986	1.06253	1.05520
17	-0.09282	-0.09770	-0.10747	-0.11480
18	-1.24572	-1.25061	-1.26038	-1.26771
19	-2.16903	-2.17391	-2.17880	-2.18124
20	-2.70151	-2.70396	-2.70640	-2.70640

INITIAL CHANNEL: 1
 FINAL CHANNEL: 4
 COORD. NUMBER: 911001

TOTAL NO. SAMPLES: 1024
 SCAN PERIOD (SEC.): 0.3333E-02
 REFERENCE FREQ (HZ): 0.2000E 02

DATA PTS./CH., USED: 255 AVAIL.: 256

SIGNAL BIAS AND R.M.S. VALUES

CHANNEL	BIAS	R.M.S.
1	0.00445	2.0098
2	0.00444	2.0098
3	0.00444	2.0098
4	0.00442	2.0099

FOURIER COEFFICIENTS FOR HARMONIC 1

CHANNEL	COS	SIN	PHASE	MAG
1	1.8424	-2.1585	49.52	2.8379
2	1.8367	-2.1634	49.67	2.8379
3	1.8308	-2.1685	49.83	2.8380
4	1.8249	-2.1735	49.98	2.8381

FOURIER COEFFICIENTS FOR HARMONIC 2

CHANNEL	COS	SIN	PHASE	MAG
1	-0.0034	0.0091	-110.30	0.0097
2	-0.0031	0.0091	-108.86	0.0097
3	-0.0030	0.0092	-108.40	0.0096
4	-0.0031	0.0088	-109.56	0.0093

DATA 4

1024 DATA POINTS

SCAN RATE 3000 HERTZ

COORDINATION NUMBER 911002

1	-1.93698	-1.99804	-2.05178	-2.10308
2	-2.59648	-2.62335	-2.65022	-2.67465
3	-2.80166	-2.79922	-2.79433	-2.78700
4	-2.52565	-2.49389	-2.45725	-2.42062
5	-1.80019	-1.73913	-1.68051	-1.61700
6	-0.74499	-0.67171	-0.59599	-0.51783
7	0.42990	0.50562	0.58378	0.65950
8	1.54861	1.61456	1.68051	1.73913
9	2.40107	2.44260	2.48168	2.51832
10	2.83097	2.84074	2.85051	2.85540
11	2.78945	2.76991	2.75037	2.72594
12	2.28139	2.23253	2.18124	2.12750
13	1.37763	1.30923	1.23840	1.16756
14	0.23449	0.15633	0.08061	0.00244
15	-0.93307	-1.00879	-1.07963	-1.15535
16	-1.94187	-1.99804	-2.05178	-2.10308
17	-2.59892	-2.62579	-2.65266	-2.67709
18	-2.80166	-2.79922	-2.79433	-2.78700
19	-2.52076	-2.49145	-2.45481	-2.41573
20	-1.79531	-1.73669	-1.67318	-1.60967

INITIAL CHANNEL: 1
 FINAL CHANNEL: 4
 COORD. NUMBER: 911002

TOTAL NO. SAMPLES: 1024
 SCAN PERIOD (SEC.): 0.3333E-03
 REFERENCE FREQ (HZ): 0.2000E 03

DATA PTS./CH., USED: 255 AVAIL.: 256

SIGNAL BIAS AND R.M.S. VALUES

CHANNEL	BIAS	R.M.S.
1	0.0369	2.0113
2	0.0362	2.0116
3	0.0357	2.0114
4	0.0353	2.0112

FOURIER COEFFICIENTS FOR HARMONIC 1

CHANNEL	COS	SIN	PHASE	MAG
1	-1.0177	-2.6559	110.97	2.8442
2	-1.0901	-2.6274	112.53	2.8445
3	-1.1613	-2.5964	114.10	2.8443
4	-1.2314	-2.5636	115.66	2.8440

FOURIER COEFFICIENTS FOR HARMONIC 2

CHANNEL	COS	SIN	PHASE	MAG
1	0.0030	0.0011	-19.56	0.0032
2	0.0036	0.0009	-14.15	0.0037
3	0.0040	0.0000	-0.64	0.0040
4	0.0041	-0.0009	12.82	0.0042

DATA 5

1024 DATA POINTS

SCAN RATE 10000 HERTZ

COORDINATION NUMBER 911003

1	-1.21641	-1.59013	-1.89301	-2.16658
2	-2.48656	-2.63801	-2.74792	-2.80410
3	-2.78700	-2.70640	-2.57694	-2.40596
4	-2.02247	-1.73669	-1.41426	-1.06253
5	-0.45921	-0.07084	0.31265	0.69370
6	1.26236	1.62677	1.92721	2.19834
7	2.54274	2.69907	2.80410	2.85295
8	2.84074	2.76258	2.63556	2.46214
9	2.08109	1.79775	1.47289	1.12115
10	0.52760	0.14411	-0.23449	-0.62531
11	-1.21397	-1.55349	-1.85882	-2.13727
12	-2.48412	-2.63556	-2.74792	-2.80166
13	-2.78700	-2.70884	-2.57938	-2.40840
14	-2.02491	-1.73913	-1.41671	-1.06497
15	-0.46165	-0.07816	0.30288	0.68637
16	1.27992	1.61700	1.92233	2.19101
17	2.54274	2.69663	2.80410	2.85540
18	2.84074	2.76258	2.63556	2.46458
19	2.08598	1.79775	1.47777	1.12848
20	0.53493	0.13679	-0.24426	-0.62531

INITIAL CHANNEL: 1
 FINAL CHANNEL: 4
 COORD. NUMBER: 911003

TOTAL NO. SAMPLES: 1024
 SCAN PERIOD (SEC.): 0.1000E-03
 REFERENCE FREQ (HZ): 0.1000E 04

DATA PTS./CH., USED: 250 AVAIL.: 256

SIGNAL BIAS AND R.M.S. VALUES

CHANNEL	BIAS	R.M.S.
1	0.0313	2.0130
2	0.0318	2.0129
3	0.0319	2.0130
4	0.0316	2.0127

FOURIER COEFFICIENTS FOR HARMONIC 1

CHANNEL	COS	SIN	PHASE	MAG
1	0.5409	-2.7947	79.05	2.8465
2	0.1554	-2.8422	86.87	2.8464
3	-0.2294	-2.8373	94.62	2.8465
4	-0.6140	-2.7792	102.46	2.8462

FOURIER COEFFICIENTS FOR HARMONIC 2

CHANNEL	COS	SIN	PHASE	MAG
1	0.0040	0.0017	-22.64	0.0044
2	0.0029	0.0005	-9.73	0.0030
3	0.0031	-0.0013	22.81	0.0033
4	0.0040	-0.0027	33.41	0.0048

LIST OF REFERENCES

1. Englehardt, C. D., Data Acquisition System for Unsteady Aerodynamic Investigation, M. S. Thesis, Naval Postgraduate School, 1977.
2. Datal Systems, Inc., Sinetrac Series Model ST-800-32S/16D A/D Peripheral Systems Instruction Manual, Document No. ST8AMH3701, 1978.
3. Intel Corporation, MDS-800 Intellec Microcomputer Development System Hardware Reference Manual, 1975.
4. Teletype Corporation, Model 40 Installation and Servicing Manual, Manual 347, Issue 3, 1976.
5. International Business Machines, Component Description: IBM 2701 Data Adapter Unit, File No. 2701-09, 1971.
6. Intel Corporation, SBC 534 Four Channel Communications Expansion Board Hardware Reference Manual, Manual No. 9800450A, 1977.
7. Intel Corporation, 8080 Microcomputer Systems User's Manual, 1975.
8. Intel Corporation, Peripheral Design Handbook, 1978.
9. Intel Corporation, 8080 Assembly Language Programming Manual, 1976.
10. Digital Research Corporation, An Introduction to CP/M Features and Facilities, 1976.
11. Casko, J. D., A Microprocessor Controlled Automatic Data Logging System (ADL), M. S. Thesis, Naval Postgraduate School, 1977.
12. Adam Osborne and Associates, Inc., An Introduction to Microcomputers, 1975.
13. Pickelsimer, B. M., Data Reduction for the Unsteady Aerodynamics on a Circulation Control Airfoil, M. S. Thesis, Naval Postgraduate School, 1977.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4. Professor L. V. Schmidt, Code 67sx Department of Aeronautics Naval Postgraduate School Monterey, California 93940	5
5. Kathryn Strutynski, Code 0141 W. R. Church Computer Center Naval Postgraduate School Monterey, California 93940	1
6. LT Mack Taylor Elliott, USN Air Anti-Submarine Squadron Forty-One Naval Air Station, North Island San Diego, California 92135	1



Thesis
E3687
c.1

Elliott
High speed data
acquisition system.

178459

Thesis
E3687
c.1

Elliott
High speed data
acquisition system.

178459

thesE3687

High speed data acquisition system.



3 2768 001 89303 5

DUDLEY KNOX LIBRARY